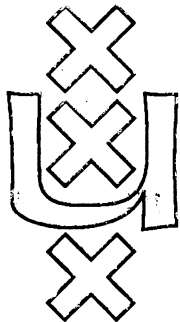


**Institute for Language, Logic and Information**

**FLEXIBLE MONTAGUE GRAMMAR**

Herman Hendriks

ITLI Prepublication Series  
for Logic, Semantics and Philosophy of Language LP-90-08



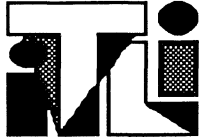
**University of Amsterdam**

## The ITLI Prepublication Series

1986

- 86-01 The Institute of Language, Logic and Information  
A Semantical Model for Integration and Modularization of Rules
- 86-02 Peter van Emde Boas  
Categorial Grammar and Lambda Calculus
- 86-03 Johan van Benthem  
A Relational Formulation of the Theory of Types
- 86-04 Reinhard Muskens  
Some Complete Logics for Branched Time, Part I Well-founded Time,  
Logical Syntax Forward looking Operators
- 86-05 Kenneth A. Bowen, Dick de Jongh  
Type shifting Rules and the Semantics of Interrogatives
- 86-06 Johan van Benthem  
Frame Representations and Discourse Representations
- 1987 87-01 Jeroen Groenendijk, Martin Stokhof  
Unique Normal Forms for Lambda Calculus with Surjective Pairing
- 87-02 Renate Bartsch  
Polyadic quantifiers
- 87-03 Jan Willem Klop, Roel de Vrijer  
Traditional Logicians and de Morgan's Example
- 87-04 Johan van Benthem  
Temporal Adverbials in the Two Track Theory of Time
- 87-05 Victor Sánchez Valencia  
Categorial Grammar and Type Theory
- 87-06 Eleonore Oversteegen  
The Construction of Properties under Perspectives
- 87-07 Johan van Benthem  
Type Change in Semantics: The Scope of Quantification and Coordination
- 87-08 Renate Bartsch
- 87-09 Herman Hendriks
- 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' Realizability
- ML-88-02 M.D.G. Swaen  
The Arithmetical Fragment of Martin Löf's Type Theories with weak  $\Sigma$ -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  
Maintaining Multiple Representations of  
Leen Torenvliet, Peter van Emde Boas  
Dynamic Data Structures
- CT-88-04 Dick de Jongh, Lex Hendriks  
Computations in Fragments of Intuitionistic Propositional Logic  
Gerard R. Renardel de Lavalette
- 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  
Multiple Representations of Dynamic Data Structures  
Leen Torenvliet, Peter van Emde Boas
- 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 Victor 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  
 $\Sigma$ -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  $\text{ID}_0 + \Omega_1$
- 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  
A Parallel Functional Implementation of Range Queries  
Leen Torenvliet, Willem G. Vree
- 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  
Honest Reductions, Completeness and  
Leen Torenvliet  
Nondeterministic 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  
Towards Functional Classification of Recursive Query Processing  
Peter van Emde Boas
- 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 SEE INSIDE BACK COVER



**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

## **FLEXIBLE MONTAGUE GRAMMAR**

Herman Hendriks  
Department of Philosophy  
University of Amsterdam

Received July 1990

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

Supported by NWO (Netherlands  
Organization for Scientific Research)  
Supported by Dyana, Project EBRA-3715



HERMAN HENDRIKS

## FLEXIBLE MONTAGUE GRAMMAR\*

### 0. INTRODUCTION

In this paper we shall argue that adoption of flexible type assignment in Montague grammar leads to a more adequate division of labour between the syntactic and semantic component. In syntax, things can be simplified to a great extent: it becomes possible to account for quantifier scope ambiguities without Montagovian rules of quantification or Cooper stores. Moreover, flexible type assignment enables us to generate semantic interpretations that cannot be straightforwardly represented within those approaches. And finally, the flexible approach is more successful in cases where quantification rules and stores lead to wrong results.

In section 1, 'The Semantics of PTQ', we shall first review several relevant properties of orthodox Montague grammar. Next, section 2, 'Arguments for flexibility', lists arguments that have been put forward in favour of a less rigid category-to-type assignment in model-theoretic semantics. The consequences of the apparently needed flexibility for the design of the grammar are treated in section 3, 'Flexibility in syntax and semantics'. We shall first consider the Lambek calculus, an influential theory of syntactic/semantic flexibility, and then present our flexible Montague grammar. Section 4, 'Applications', shows that the flexible grammar defined in section 3 not only has the wished-for advantages, but allows one to dispense with special syntactic devices for the representation of quantifier scope ambiguities. In the last section, 'Discussion: compositionality and contextuality', we pay attention to the position of the flexible grammar relative to the principles of compositionality and contextuality. The semantic properties of the proposed fragment are treated in an Appendix.

### 1. PTQ

In his paper 'The Proper Treatment of Quantification in Ordinary English' (i.e., Montague 1974b, PTQ henceforth), Richard Montague presented a grammar for a fragment of English which distinguishes itself from its generative predecessors by the presence of a semantic component where the expressions defined by the syntactic component are translated into expressions of the logical language IL (= Intensional

---

\* I would like to thank Renate Bartsch, Johan van Benthem, Paul Dekker, Jeroen Groenendijk, Theo Janssen, Reinhard Muskens and Martin Stokhof for their help, encouragement and/or criticism. The research for this paper was supported by the Netherlands Organization for Scientific Research (NWO) and ESPRIT Basic Research Action 3175, Dynamic Interpretation of Natural Language (DYANA).

Logic). These IL-expressions receive a model-theoretic interpretation in the usual way. Thus the English expressions are indirectly assigned a semantic interpretation, viz., via the interpretation of the logical expressions they are translated into.<sup>1</sup> The ‘PTQ fragment’ has since acquired a paradigmatic status within model-theoretic semantics. In this section we shall go into a number of aspects of the fragment, which will be considered with its original semantics (plus individual concepts – cf. Dowty, Wall and Peters (1981), Appendix III).

### 1.1 Categories

The set of syntactic categories in the PTQ fragment is defined inductively. There are two basic categories, S and E. S is the category of ‘sentences’, and E is an empty category in the sense that no expression belongs to it. Furthermore, there are compound categories: if A and B are categories, then A/B and A//B are also categories. Intuitively, A/B and A//B represent a category of expressions that take an expression of category B to form an expression of category A. As a consequence of this definition, we get an infinitely large set of categories at our disposal. In practice we shall only have to deal with a finite subset of the defined categories: the eight categories indicated below.

(1)	CATEGORY	DESCRIPTION	DEFINITION
	S	sentence	–
	E	–	–
	CN	common noun	S/E
	IV	intransitive verb phrase	S//E
	T	term (noun) phrase	S/IV = S/(S//E)
	Det	determiner	T/CN = (S/(S//E))/(S/E)
	TV	transitive verb phrase	IV/T = (S//E)/(S/(S//E))
	PV	propositional verb phrase	IV/S = (S//E)/S

With the exception of S and E, all categories contain lexical elements. (The term phrases [*The*<sub>0</sub>], [*The*<sub>1</sub>], [*The*<sub>2</sub>], ... are called syntactic variables.)

(2)	CN:	[CNman], [CNwoman], [CNunicorn], [CNtemperature], ...
	IV:	[IVwalk], [IVrise], ...
	T:	[TJohn], [TMary], ..., [ <i>The</i> <sub>0</sub> ], [ <i>The</i> <sub>1</sub> ], [ <i>The</i> <sub>2</sub> ], ...
	Det:	[Detthe], [Devery], [Deta], [Detone], ...
	TV:	[TVseek], [TVfind], ...
	PV:	[PVclaim that], [PVbelieve that], ...

---

<sup>1</sup> Cf. Janssen (1983) for a clear exposition of this framework; less clear, more concise: Montague (1974a).

## 1.2 Types

There is a *rigid* correspondence between syntactic categories and semantic types in PTQ. I.e., if an expression,  $\alpha$ , belongs to a certain category,  $C$ , then its IL-translation,  $\alpha'$ , is invariably of the unique type assigned to  $C$ :  $\text{TYPE}(C)$ . Put differently, the category-to-type assignment in PTQ,  $\text{TYPE}$ , is a *function*. This function is defined as follows:  $\text{TYPE}(S) = t$ ,  $\text{TYPE}(E) = e$ , and  $\text{TYPE}(A/B) = \text{TYPE}(A//B) = ((s, \text{TYPE}(B)), \text{TYPE}(A))$ . For our eight categories this amounts to:

(3) CATEGORY	TYPE(CATEGORY)
S	t
E	e
CN	((s,e),t)
IV	((s,e),t)
T	((s,((s,e),t)),t)
Det	((s,((s,e),t)),((s,((s,e),t)),t))
TV	((s,((s,((s,e),t)),t)),((s,e),t))
PV	((s,t),((s,e),t))

The above category-to-type assignment is of semantic importance, since it determines for each expression what kind of object in the model is assigned to that expression as its denotation: type  $t$  represents the set of *truth values*,  $e$  the set of *entities*, and  $s$  (which is not really a basic type in IL) the set of *possible worlds*. A type  $(c,d)$  stands for the set of *functions* from objects of type  $c$  to objects of type  $d$ . Whenever  $b$  equals  $t$ , we can identify such a function of type  $(c,b) = (c,t)$  with a *set* of objects of type  $c$ . And when the type is of form  $(c,(d,t))$ , we can identify the objects in its denotation with sets of ordered pairs  $\langle \gamma, \delta \rangle$ , where  $\gamma$  is an object of type  $c$ , and  $\delta$  is an object of type  $d$ ; hence with binary *relations* between objects of type  $a$  and objects of type  $b$ .

By stipulating that  $\text{TYPE}(S)$  equals  $t$ , we establish that the denotation of a sentence is a truth value. If there were expressions of type  $E$ , their denotation would be an entity. The denotation of common nouns and intransitive verb phrases is a set of individual concepts. (An *individual concept* is a function from possible worlds to entities.) Just imagine: an object of type  $((s,e),t)$  is a function from objects of type  $(s,e)$  to objects of type  $t$  – a function that can be identified with a set of objects of type  $(s,e)$ , i.e., a set of functions from possible worlds to entities, or a set of individual concepts. The denotation of a noun phrase is a set of properties of individual concepts. (A *property* of objects of type  $c$  is an object of type  $(s,(c,t))$ ; a function from possible worlds to sets of objects of type  $c$ .) A determiner denotes a binary relation between properties of individual concepts, a transitive verb phrase denotes a binary relation between properties of properties of individual concepts and individual concepts, and a propositional verb phrase denotes a

binary relation between propositions and individual concepts. (A *proposition* is a function from possible worlds to truth values.)

The rigid correspondence between categories and types has some consequences. Note, for instance, that a straightforward option for the denotation of a proper name like *John*, namely the entity John (to be denoted by an individual constant  $j$  of type  $e$  in IL), is excluded. For the category *John* belongs to (the category of noun phrases) is associated with type  $((s,((s,e),t)),t)$  by the TYPE function, and not with  $e$ . Therefore, we are obliged to translate *John* into an expression of the former type. Montague chooses  $\lambda P.\forall P(\wedge j)$ , an expression referring to the set of properties of individual concepts such that its extension contains the intension of the denotation of  $j$  as its element, and this solution turns out to work. Montague's solution is practically inevitable under the assumptions of the PTQ framework. One could, e.g., think of a different category assignment to proper names (E comes to mind), but this fails to do justice to the reality of syntax: proper names and compound noun phrases like *a woman* have a very similar distribution; they can be coordinated (*John and a woman*); etc. Such an assignment would, moreover, complicate the grammar to a great extent: every rule which applies to both proper names and compound noun phrases would have to be doubled.

### 1.3 Translation rules

The rules of the PTQ grammar consist of a syntactic and a semantic half. The syntactic rules give a recursive definition of the set of expressions that constitute the fragment of English. We shall ignore the details of syntactic analysis here.

The semantic translation rules determine (i) what is the interpretation (strictly speaking: the IL-translation) of the lexical expressions; and (ii) what is the interpretation of compound expressions, *given the interpretations of the immediate constituent expressions*. Let us look at some examples.

#### 1.3.1 Lexical expressions.

PTQ translates common nouns into non-logical constants of type  $((s,e),t)$ . The translation of *temperature*, for instance, is TEMPERATURE. This IL expression denotes some set of individual concepts. Translations like these appear to be adequate for common nouns like *temperature*, *percentage*, etc. (cf. Janssen 1984 and Gamut 1982, section 7.4.1, for arguments), but not for 'normal' common nouns like *man*, *woman*, *unicorn* which denote sets of individuals. That is why Montague supplements his translations of these common nouns as non-logical constants of type  $((s,e),t)$  with a meaning postulate (MP 2 in PTQ). We opt here for a slightly different approach, and incorporate the meaning postulates in the lexical translations (ignoring Montague's \*-notation):



- (4)
- |                          |  |
|--------------------------|--|
| [CN <i>man</i> ]         | $\lambda x.[\text{MAN}(\forall x) \wedge \exists u(x=\wedge u)]$     |
| [CN <i>woman</i> ]       | $\lambda x.[\text{WOMAN}(\forall x) \wedge \exists u(x=\wedge u)]$   |
| [CN <i>unicorn</i> ]     | $\lambda x.[\text{UNICORN}(\forall x) \wedge \exists u(x=\wedge u)]$ |
| [CN <i>temperature</i> ] | TEMPERATURE  |

In doing so, we account for the fact that these ‘normal’ common nouns denote sets of *constant* individual concepts (that is, ‘in fact’ sets of individuals). The remaining lexical expressions get the following translations:

- (5)
- |                            |  |
|----------------------------|--|
| [TV <i>walk</i> ]          | $\lambda x.\text{WALK}(\forall x)$   |
| [TV <i>rise</i> ]          | RISE   |
| [T <i>John</i> ]           | $\lambda P.\forall P(\wedge j)$  |
| [T <i>Mary</i> ]           | $\lambda P.\forall P(\wedge m)$  |
| [T <i>he<sub>i</sub></i> ] | $\lambda P.\forall P(x_i)$   |
| [Det <i>a</i> ]            | $\lambda P\lambda Q.\exists x[\forall P(x)\wedge\forall Q(x)]$                               |
| [Det <i>every</i> ]        | $\lambda P\lambda Q.\forall x[\forall P(x)\rightarrow\forall Q(x)]$                          |
| [Det <i>the</i> ]          | $\lambda P\lambda Q.\exists x[\forall y[\forall P(y)\leftrightarrow x=y]\wedge\forall Q(x)]$ |
| [Det <i>one</i> ]          | $\lambda P\lambda Q.\exists x\forall y[[\forall P(y)\wedge\forall Q(y)]\leftrightarrow x=y]$ |
| [TV <i>seek</i> ]          | $\lambda T\lambda x.\text{SEEK}(T)(\forall x)$   |
| [TV <i>find</i> ]          | $\lambda T\lambda x.\forall T(\wedge\lambda y.\text{FIND}(\forall y)(\forall x))$            |
| [PV <i>claim that</i> ]    | $\lambda p\lambda x.\text{CLAIM}(p)(\forall x)$  |
| [PV <i>believe that</i> ]  | $\lambda p\lambda x.\text{BELIEVE}(p)(\forall x)$  |

Montague’s meaning postulate 3 is hidden in the translation of *walk*. This intransitive verb holds of the *value* of the individual concepts that constitute its denotation: the subject position of verbs like these is extensional. The meaning postulate does not hold for *rise*; and so this verb is simply translated as a non-logical constant of type ((s,e),t). In the translations of *seek*, *find*, *claim that* and *believe that* we have incorporated the PTQ meaning postulates 5 and 6, respectively, accounting for the extensionality of the subject position of TVs and PVs. In the case of *find* the object position is extensional as well, which explains why the translation of this verb also shows traces of Montague’s meaning postulate 4.

### 1.3.2 Compound expressions

PTQ contains a number of rules for the translation of compound expressions that belong to the categories mentioned in section 1.1 above. (Because we do not consider relative clauses, the rule for the formation of compound CNs is omitted here. The CN rule of quantification is absent for the same reason.)  $\alpha'$  and  $\beta'$  denote the translations of  $\alpha$  and  $\beta$ , respectively.

- (A) Rules of application
- (I)  $\alpha_T + \beta_{IV} \rightarrow [{}_S \alpha_T \beta_{IV}]$  Translation:  $\alpha'(\wedge\beta')$
- (II)  $\alpha_{TV} + \beta_T \rightarrow [{}_{IV} \alpha_{TV} \beta_T]$  Translation:  $\alpha'(\wedge\beta')$
- (III)  $\alpha_{Det} + \beta_{CN} \rightarrow [{}_T \alpha_{Det} \beta_{CN}]$  Translation:  $\alpha'(\wedge\beta')$
- (IV)  $\alpha_{PV} + \beta_S \rightarrow [{}_{IV} \alpha_{PV} \beta_S]$  Translation:  $\alpha'(\wedge\beta')$
- (B) Rules of conjunction and disjunction.
- (V)  $\alpha_S + \beta_S \rightarrow [{}_S \alpha_S \text{ and } \beta_S]$  Translation:  $[ \alpha' \wedge \beta' ]$
- (VI)  $\alpha_S + \beta_S \rightarrow [{}_S \alpha_S \text{ or } \beta_S]$  Translation:  $[ \alpha' \vee \beta' ]$
- (VII)  $\alpha_{IV} + \beta_{IV} \rightarrow [{}_{IV} \alpha_{IV} \text{ and } \beta_{IV}]$  Translation:  $\lambda x. [ \alpha'(x) \wedge \beta'(x) ]$
- (VIII)  $\alpha_{IV} + \beta_{IV} \rightarrow [{}_{IV} \alpha_{IV} \text{ or } \beta_{IV}]$  Translation:  $\lambda x. [ \alpha'(x) \vee \beta'(x) ]$
- (IX)  $\alpha_T + \beta_T \rightarrow [{}_T \alpha_T \text{ and } \beta_T]$  Translation:  $\lambda P. [ \alpha'(P) \wedge \beta'(P) ]$
- (X)  $\alpha_T + \beta_T \rightarrow [{}_T \alpha_T \text{ or } \beta_T]$  Translation:  $\lambda P. [ \alpha'(P) \vee \beta'(P) ]$
- (C) Rules of quantification (for  $n \in \{0,1,2,\dots\}$ ).
- (XI), $n$   $\alpha_T + \beta_S \rightarrow \beta_S^*$  Translation:  $\alpha'(\wedge\lambda x_n.\beta')$
- (XII), $n$   $\alpha_T + \beta_{IV} \rightarrow \beta_{IV}^*$  Translation:  $\lambda y.\alpha'(\wedge\lambda x_n.[\beta'(y)])$

In the rule schemes (XI), $n$  and (XII), $n$ ,  $\beta_S^*$  and  $\beta_{IV}^*$  stand for  $\beta_S$  and  $\beta_{IV}$ , respectively, with (i) if  $\alpha_T$  is a syntactic variable, all occurrences of  $he_n$  replaced by  $\alpha_T$ ; and (ii) if  $\alpha_T$  is not a syntactic variable, the first occurrence of  $he_n$  replaced by  $\alpha_T$ , and the remaining occurrences of  $he_n$  replaced by anaphoric pronouns of appropriate case and gender.

### 1.3.3 Scope ambiguities

The rules (XI), $n$  and (XII), $n$  are used to represent quantifier scope and *de dicto/de re* ambiguities. E.g., in constructing the sentence *Every man finds a woman* we can only apply the rules of application (I), (II) and (III), which yields the wide scope-*every* translation (6). But we can also construct *Every man finds him<sub>8</sub>* first, which translates as (7), and then apply (XI),8 (an instantiation of the rule scheme (XII), $n$ ) to *a woman* and *Every man finds him<sub>8</sub>*. In this way we get the wide scope-*a* translation (8).

- (6)  $\forall u[MAN(u) \rightarrow \exists v[WOMAN(v) \wedge FIND(v)(u)]]$
- (7)  $\lambda x_8.\forall u[MAN(u) \rightarrow FIND(\wedge x_8)(u)]$
- (8)  $\exists v[WOMAN(v) \wedge \forall u[MAN(u) \rightarrow FIND(v)(u)]]$

Similarly, sentence (9) is assigned three readings by the above semantics.

- (9)  $[{}_S \text{Mary}_T [{}_{IV} \text{walks}_{IV} \text{ and } [{}_{IV} \text{seeks}_{TV} [{}_T \text{every}_{Det} \text{unicorn}_{CN} ]_T ]_{IV} ]_S]$

The *de dicto* reading (10) can be built up without rules of quantification. Apart from conjunction rule (VII), only rules of applications are used in its construction.

$$(10) \quad \text{WALK}(m) \wedge \text{SEEK}(\wedge \lambda Q. \forall v[\text{UNICORN}(v) \rightarrow \forall Q(\wedge v)])(m)$$

Within the PTQ framework, sentence (9) is assigned two more translations, viz., (11) and (12), in which the quantified noun phrase *every unicorn* is read *de re*. In order to get these translations, we have to make use of the rule schemes of quantification (XII),n and (XI),n, respectively.

$$(11) \quad \text{WALK}(m) \wedge \forall v[\text{UNICORN}(v) \rightarrow \text{SEEK}(\wedge \lambda P. \forall P(v))(m)]$$

$$(12) \quad \forall v[\text{UNICORN}(v) \rightarrow [\text{WALK}(m) \wedge \text{SEEK}(\wedge \lambda P. \forall P(v))(m)]]$$

Both translations require that we first build up the verb phrase *seek him<sub>i</sub>*, with some syntactic variable occupying the position of the quantified direct object.

To get (11), we first apply (XII),i, quantifying *every unicorn* into the verb phrase *seek him<sub>i</sub>*, with the result *seek every unicorn*. Then (VII) is used to form *walk and seek every unicorn*, which is finally combined with *Mary*.

To get (12), we first conjoin *seek him<sub>i</sub>* with *walk*; then we use (I), which gets us the sentence *Mary walks and seeks him<sub>i</sub>*; and finally we apply (XI),i, quantifying-in *every unicorn* into this sentence.

Note, however, that whereas (11) is okay, (12) is an incorrect translation of sentence (9), witness the equivalence of the following formulas:

$$(13) \quad \begin{aligned} & \forall v[\text{UNICORN}(v) \rightarrow [\text{WALK}(m) \wedge \text{SEEK}(\wedge \lambda P. \forall P(v))(m)]] \Leftrightarrow \\ & \forall v[[\text{UNICORN}(v) \rightarrow \text{WALK}(m)] \wedge [\text{UNICORN}(v) \rightarrow \text{SEEK}(\wedge \lambda P. \forall P(v))(m)]] \Leftrightarrow \\ & \forall v[\text{UNICORN}(v) \rightarrow \text{WALK}(m)] \wedge \forall v[\text{UNICORN}(v) \rightarrow \text{SEEK}(\wedge \lambda P. \forall P(v))(m)] \Leftrightarrow \\ & [\exists v \text{UNICORN}(v) \rightarrow \text{WALK}(m)] \wedge \forall v[\text{UNICORN}(v) \rightarrow \text{SEEK}(\wedge \lambda P. \forall P(v))(m)] \end{aligned}$$

(12) apparently means: Mary walks if there are unicorns, and for every unicorn it holds that Mary seeks it. A truth in every possible world without unicorns, whether or not Mary walks. This is wrong: sentence (9) does not express that Mary's walking is dependent on the existence of unicorns.

An analogous problem arises with (14).<sup>2</sup> This sentence is assigned two readings. The correct reading (15) is obtained without rules of quantification. But using (XI),n, we arrive at the obvious non-reading (16).

$$(14) \quad \text{John runs and no unicorn walks}$$

$$(15) \quad \text{RUN}(j) \wedge \neg \exists v[\text{UNICORN}(v) \wedge \text{WALK}(v)]$$

$$(16) \quad \begin{aligned} & \neg \exists v[\text{UNICORN}(v) \wedge \text{RUN}(j) \wedge \text{WALK}(v)] \Leftrightarrow \\ & \neg \text{RUN}(j) \vee \neg \exists v[\text{UNICORN}(v) \wedge \text{WALK}(v)] \end{aligned}$$

---

<sup>2</sup> Paul Dekker (p.c.).

In the sequel we shall see that the flexible Montague grammar defined in section 3 assigns only the correct readings in these cases.

## 2. ARGUMENTS FOR FLEXIBILITY

We saw in section 1.1 above that, due to the rigid category-to-type assignment in PTQ, a proper name like *John* is not translated as a non-logical constant of type  $e$ , but as a complex IL-expression of type  $((s,((s,e),t)),t): \lambda P.\forall P(^j)$ . This, in its turn, has further consequences: an extensional transitive verb like *find*, for example, is translated as  $\lambda T\lambda x.\forall T(^{\lambda y.FIND(\forall y)(\forall x)})$ , of type  $TYPE(TV) = ((s,((s,((s,e),t)),t)),((s,e),t))$ , and not as  $\lambda y\lambda x.FIND(\forall y)(\forall x)$  of type  $((s,e),((s,e),t))$ , which would be the most direct way to obtain the translation  $FIND(m)(j)$  for *John finds Mary* under the assumptions that (i) proper names have denotations of type  $e$ , and that (ii) the rules of application (I) and (II) translate as  $\alpha'(^{\beta'})$ . The rigid category-to-type assignment entails a considerable distortion of the semantics, a distortion which, for that matter, has not yet reached its full extent with PTQ, as Bach (1980) argues. For instance, there are verbs whose subject position is intensional, cf. (17) and (18).

(17) A man is missing

(18) A unicorn seems to be approaching

In  $BE\text{-}MISSING(^{\lambda Q.\exists v[MAN(v)\wedge^{\forall}Q(v)])}$ , the non-referential reading of sentence (17), the translation of the intransitive verb phrase,  $BE\text{-}MISSING$ , is not of  $TYPE(IV) = ((s,e),t)$ , but of type  $((s,TYPE(NP)),t) = ((s,((s,((s,e),t)),t)),t)$ . *Mutatis mutandis*, the same holds for (18). The existence of intransitive verb phrases like *be missing* entails within the PTQ framework a raising of the type assignment (plus a change in the definition) of a number of categories: for *all* intransitive verbs (hence also for extensional ones like *walk*)  $TYPE(IV)$  should be changed into  $((s,((s,((s,e),t)),t)),t)$ ; without exception, *all* transitive verbs must be assigned a new  $TYPE(TV)$ :  $((s,((s,((s,e),t)),t)),((s,((s,((s,e),t)),t)),t))$  (while no transitive verb with this amount of intensionality is known, cf. Partee and Rooth: ‘there are [apparently] no basic lexical verbs that are intensional with respect to both subject and object’ (1983, p. 379)); and the new  $TYPE(PV)$  for *all* PVs becomes  $((s,t),((s,((s,((s,e),t)),t)),t))$ . The resulting category-to-type assignment is the one found in Montague’s ‘Universal grammar’ (= UG, Montague (1974a)), plus individual concepts.

In itself, such a complication is not a disaster, provided that it serves the empirical adequacy of the theory. But there are reasons to assume that Montague’s strategy of ‘generalizing to the worst case’ – uniformly assign *all* members of a certain syntactic category the ‘highest’ type that is needed for *some* expression in that category – is not the one to be pursued. This has been claimed by Van Benthem (1986), Groenendijk and

Stokhof (1984, 1987), Keenan and Faltz (1985), Partee (1986), and Partee and Rooth (1983), among others. The point is best illustrated with a sentence like (19).

(19) John seeks a fish or a bike

(20)  $\text{SEEK}(\wedge\lambda P.[\exists v[\text{FISH}(v)\wedge\vee P(\wedge v)] \vee \exists v[\text{BIKE}(v)\wedge\vee P(\wedge v)]](j)$

(21)  $\exists v[\text{FISH}(v)\wedge\text{SEEK}(\wedge\lambda P.\vee P(\wedge v))(j)] \vee \exists v[\text{BIKE}(v)\wedge\text{SEEK}(\wedge\lambda P.\vee P(\wedge v))(j)]$

This sentence has a *de dicto* and a *de re* reading to start with. The *de dicto* reading, (20), can be accounted for in PTQ by applying rule (X) to *a fish* and *a bike: a fish or a bike*, with the reduced translation (22):

(22)  $\lambda P.[\exists v[\text{FISH}(v)\wedge\vee P(\wedge v)] \vee \exists v[\text{BIKE}(v)\wedge\vee P(\wedge v)]]$

If we subsequently combine (22) with the translations of *seek* (rule (II)) and *John* (rule (I)), the result reduces to (20). (21) is also among the possibilities of PTQ. For this reading we first build up *seek him<sub>i</sub>* (with rule (II)). Rule (I) combines this intransitive verb phrase with the term *John*. The resulting ‘sentence’ is *John seeks him<sub>i</sub>*. With rule (XI)<sub>i</sub> we then quantify the disjunctive noun phrase *a fish or a bike* into *John seeks him<sub>i</sub>*:

(23)  $\lambda P.[\exists v[\text{FISH}(v)\wedge\vee P(\wedge v)] \vee \exists v[\text{BIKE}(v)\wedge\vee P(\wedge v)]](\wedge\lambda x_i.\text{SEEK}(\wedge\lambda P.\vee P(x_i))(j))$

(23)  $\Leftrightarrow$  (21). But (20) and (21) do not exhaust the possibilities of (19). As Partee and Rooth show, there is yet a third reading of the sentence: the so-called *de dicto* wide scope-*or* reading, which is suggested by the continuation ‘... but I don’t know which’, and which can be formulated in IL as:

(24)  $\text{SEEK}(\wedge\lambda P.\exists v[\text{FISH}(v)\wedge\vee P(\wedge v)])(j) \vee \text{SEEK}(\wedge\lambda P.\exists v[\text{BIKE}(v)\wedge\vee P(\wedge v)])(j)$

PTQ does not account for (24). Partee and Rooth indicate how this reading could be constructed (we slightly adapt their types): suppose that every expression  $\gamma$  with translation  $\gamma'$  of type  $c$  also has a translation of type  $((s,((s,c),b)),b)$ :  $\lambda Q(s,((s,c),b)).\vee Q(\wedge\gamma')$ , then [with  $b = ((s,e),t)$ ] *a fish* and *a bike* have translations of type  $((s,((s,((s,((s,e),t)),t)),((s,e),t))),((s,e),t))$ :

(25)  $\lambda Q_{((s,((s,((s,((s,e),t)),t)),((s,e),t))),((s,e),t))}.\vee Q(\wedge\lambda P.\exists v[\text{FISH}(v)\wedge\vee P(\wedge v)])$  and  
 $\lambda Q_{((s,((s,((s,((s,e),t)),t)),((s,e),t))),((s,e),t))}.\vee Q(\wedge\lambda P.\exists v[\text{BIKE}(v)\wedge\vee P(\wedge v)]);$

and suppose, moreover, that the combination of these expressions into *a fish or a bike* follows the general syntactic/semantic scheme of *generalized disjunction*<sup>3</sup>, then *a fish or a bike* is assigned translation (26) of type  $((s,((s,((s,((s,e),t)),t)),((s,e),t))),((s,e),t))$ .

<sup>3</sup> Defined in Gazdar (1980), Keenan and Faltz (1985), Partee and Rooth (1983); cf. below, section 3.2.1.2.

$$(26) \quad \lambda Q \lambda y. [\vee Q(\wedge \lambda P. \exists v [FISH(v) \wedge \vee P(\wedge v)])(y) \vee \vee Q(\wedge \lambda P. \exists v [BIKE(v) \wedge \vee P(\wedge v)])(y)]$$

The application of (26) to  $\wedge \lambda T \lambda x. SEEK(T)(\vee x)$  reduces to (27). And the application of  $\lambda P. \vee P(\wedge j)$  to the intension of (27) yields (24).

$$(27) \quad \lambda y. [SEEK(\wedge \lambda P. \exists v [FISH(v) \wedge \vee P(\wedge v)])(\vee y) \vee SEEK(\wedge \lambda P. \exists v [BIKE(v) \wedge \vee P(\wedge v)])(\vee y)]$$

In consideration of such examples, Partee and Rooth conclude that we should give up Montague's strategy – uniformly assign *all* members of a certain syntactic category the 'highest' type that is needed for *some* expression in that category. Apparently, there is not always a 'highest' type available, or a worst case to generalize to: in the case of the *de dicto* wide scope-*or* reading of (19) for example, we need to have the type  $((s, ((s, ((s, ((s, e), t)), t)), ((s, e), t))), ((s, e), t))$  for noun phrases; and for the *de dicto* wide scope-*or* readings of the sentences in (28) that express an uncertainty of the speaker (cf. the continuation '... but I don't know exactly what she claims')<sup>4</sup>, we need to assign ever higher types to the quantified noun phrase *a fish or a bike*.

- (28) Wanda claims that John seeks a fish or a bike  
 Wanda claims that Jim knows that John seeks a fish or a bike  
 Wanda claims that Jules knows that Jim knows that John seeks a fish or a bike  
 Wanda claims that Jim knows that Jules knows that Jim knows that John seeks a fish or a bike  
 Wanda claims that Jules knows that Jim knows that Jules knows that Jim knows that John seeks a fish or a bike  
 ...

Partee and Rooth propose a 'reverse' strategy instead: abandon the uniform and rigid category-to-type assignment; give every expression in the lexicon a basic translation of the minimal type available for that expression (extensional transitive verbs like *find* and intensional ones like *seek* thus get different minimal types; the lexical type of the term *John* is 'lower' than the lexical type of *a woman*); and derive the translations of higher types from the lexical translation by general rules. (A different argumentation for this 'reverse' strategy is given by Groenendijk and Stokhof (1984, 1987). They show that certain intuitive entailment relations exist in lower types, but are absent from the level of the 'highest type', which is also needed.) This 'reverse' strategy opens up interesting perspectives: cf. sentence (29) with its two readings (30) and (31).

- (29) John caught and ate a fish  
 (30)  $\exists v [FISH(v) \wedge CATCH(v)(j)] \wedge \exists v [FISH(v) \wedge EAT(v)(j)]$   
 (31)  $\exists v [FISH(v) \wedge CATCH(v)(j) \wedge EAT(v)(j)]$

<sup>4</sup> I.e., the readings in which a disjunction of claims is ascribed to Wanda.

When PTQ is enriched with a rule of generalized conjunction for transitive verbs,

$$(32) \quad \alpha_{TV} + \beta_{TV} \rightarrow [_{TV} \alpha_{TV} \text{ and } \beta_{TV}] \quad \lambda y \lambda x. [\alpha'(y)(x) \wedge \beta'(y)(x)],$$

we can reach reading (30) by using this rule to make *caught and ate*, and combining this further, via rule (II) and (I), with *a fish* and *John*, respectively. Reading (31) comes about by quantifying the noun phrase *a fish* into the sentence *John caught and ate him<sub>i</sub>*, using rule (XI),i.

However, suppose that the transitive verbs *catch* and *eat* have lexical translations

$$(33) \quad \lambda y \lambda x. \text{CATCH}(\forall y)(\forall x) \text{ and } \lambda y \lambda x. \text{EAT}(\forall y)(\forall x),$$

of the minimal type  $((s,e),((s,e),t))$ ; and that the translations of the higher type  $((s,((s,((s,e),t)),t)),((s,e),t))$  which PTQ assigns to these expressions,

$$(34) \quad \lambda T \lambda x. \forall T(\wedge \lambda y. \text{CATCH}(\forall y)(\forall x)) \text{ and } \lambda T \lambda x. \forall T(\wedge \lambda y. \text{EAT}(\forall y)(\forall x)),$$

are derivable from the lexical translations. In that case we can generate both readings without using rules of quantification or syntactic variables. (30) is the result of conjoining the derived translations in (34), (and applying the conjunction to direct object and subject):

$$(35) \quad \lambda T \lambda x. [\forall T(\wedge \lambda y. \text{CATCH}(\forall y)(\forall x)) \wedge \forall T(\wedge \lambda y. \text{EAT}(\forall y)(\forall x))]$$

And (31) is reached by conjoining the basic translations in (33):

$$(36) \quad \lambda y \lambda x. [\text{CATCH}(\forall y)(\forall x) \wedge \text{EAT}(\forall y)(\forall x)],$$

followed by an application of the same rule that also derives (34) from (33), with the following result (which is then applied to direct object and subject):

$$(37) \quad \lambda T \lambda x. \forall T(\wedge \lambda y. [\text{CATCH}(\forall y)(\forall x) \wedge \text{EAT}(\forall y)(\forall x)]),$$

This suggests that we can put the ‘reverse’ strategy to use in an account of scope ambiguities without syntactic variables or rules of quantification.

Rules of quantification are syntactic rules which have a primarily semantic motivation. They also serve a more syntactic function (the representation of anaphoric pronouns), but the main reason for their introduction is semantic: the representation of quantifier scope ambiguities. From a syntactic point of view this can be seen as an exceptional interference in syntax by the semantic principle of compositionality<sup>5</sup>. Another

---

<sup>5</sup> Rules of quantification are the only rules in PTQ that disturb the ‘intuitive’ syntactic constituent structure.

reason for seeking alternatives is the fact that the syntactic operation of these rules ('if  $\alpha_T$  is a syntactic variable, replace all occurrences of  $he_n$  by  $\alpha_T$ ; and if  $\alpha_T$  is not a syntactic variable, replace the first occurrence of  $he_n$  by  $\alpha_T$ , and the remaining occurrences of  $he_n$  by anaphoric pronouns of appropriate case and gender') is somewhat strange, to put it mildly. In most theories of syntax (including the ones which are attractive from the point of view of compositional semantics, like GPSG<sup>6</sup> or flexible categorial grammar<sup>7</sup>), such syntactic operations are simply not allowed. But also for those who do not feel bound to restrictions on possible syntactic operations, there are reasons to look for alternatives: we saw that they provide a sentence like (9) with reading (12), and Janssen (1983) amply discusses the unwanted consequences of using syntactic variables and rules of quantification (the production of 'sentences' like  $he_{38}$  *walks* is one of the more innocent among them), and introduces a metagrammatical 'variable principle' in order to control things.

Therefore, in the more or less recent past various attempts have been made to avoid these syntactic devices, while preserving the results of PTQ as regards the representation of quantifier scope ambiguities. However, the main alternative of 'Cooper storage' (see Cooper (1983)) met with objections from a semantic viewpoint (Landman and Moerdijk (1983), Janssen (1983)). The present paper can be considered the umpteenth attempt. It must be kept in mind that we shall not deal with the second function of rules of quantification, the representation of anaphoric pronouns. The theory of pronouns in PTQ is syntactically hardly explicit (for instance, it does not distinguish between reflexive and personal pronouns). A successful attempt to make it less implicit has been made by Landman and Moerdijk (1983). But this theory treats anaphoric pronouns without exception as bound variables in the semantics, just as PTQ does. Reinhart (1983), Kamp (1981), and many others since, have shown that not every occurrence of an anaphoric pronoun can be considered as such: the true theory of anaphoric pronouns will at least have to include a 'discourse representation theory'. Nonetheless, part of the intrasentential behaviour of terms (including all occurrences of reflexive pronouns) does have to be analyzed in terms of bound variables. Here, for instance, Pollard and Sag's (1983) theory of reflexives and reciprocals constitutes an alternative which is consistent with the proposals made in the present paper.

### 3. FLEXIBILITY IN SYNTAX AND SEMANTICS

In this section we shall present a formal elaboration of the 'reverse' strategy sketched above. The traditional correspondence between categories and types (which uniformly assigns every expression in a certain category a translation of the 'highest' type that is needed for some expression in that category) is given up. Instead, we give every

---

<sup>6</sup> See Gazdar, Klein, Pullum and Sag (1985); Klein and Sag (1985).

<sup>7</sup> See Lambek (1958); Van Benthem (1986, chapter 7); Moortgat (1988).



expression a translation of its minimal type, and derive the translations of higher (rather: other) types from this lexical translation by general rules, thus obtaining the flexible Montague grammar of section 3.2. But before we have reached that point, we shall, in section 3.1, first explore the Lambek calculus, a flexible theory of syntax which has been provided with a semantics by Van Benthem (1986, chapter 7). It will turn out that this theory is either too weak (in its directional version) or too strong (in its non-directional version) for our purposes. The flexible Montague grammar of section 3.2 is essentially an intensional version of a subsystem of the non-directional Lambek calculus, together with a semantics along the lines of Van Benthem.

### 3.1 Lambek calculus

In its original (directional) version (Lambek (1958)), the Lambek calculus  $L$  is a theory of syntactic categories. In keeping with the lexicalist tradition of categorial grammar, the combinatorial properties of expressions are determined by the categories they belong to. Restricting ourselves to the product-free part of  $L$ , we distinguish atomic categories and two kinds of compound categories: left- ( $y \setminus x$ ) and right-searching ( $x / y$ ) functors:

- (38) Let  $ATOM$  be a finite set of atomic categories. Then  $CAT$ , the set of categories, is the smallest set such that (i)  $ATOM \subseteq CAT$ , and (ii) if  $x \in CAT$  and  $y \in CAT$ , then  $x / y \in CAT$  and  $y \setminus x \in CAT$ .

$L$  consists of a set of axioms and five inference rules. It characterizes the derivability of so-called *sequents* of categories, cf. (39). In (40) and (41),  $x, y, z$  denote categories and  $T, U, V$  finite sequences of categories ( $T$  non-empty).

- (39) A *sequent* is a pair  $\langle T, x \rangle$ , where  $T$  is a finite non-empty sequence of categories ( $T = \langle x_1, \dots, x_n \rangle$ , with  $n \geq 1$ ) and  $x$  is a category.

$T$  is called the *left-hand side*, and  $x$  the *right-hand side* of  $\langle T, x \rangle$ .

For the sequent  $\langle \langle x_1, \dots, x_n \rangle, x \rangle$  we write  $x_1, \dots, x_n \Rightarrow x$ .

- (40) The *axioms* of  $L$  are all the sequents of the form  $x \Rightarrow x$ .

- (41) The *inference rules* of  $L$  are  $/L$ ,  $\setminus L$ ,  $/R$ ,  $\setminus R$  and Cut:

$$\begin{array}{l}
 /L \quad \frac{T \Rightarrow y \quad U, x, V \Rightarrow z}{U, x / y, T, V \Rightarrow z} \\
 \setminus L \quad \frac{T \Rightarrow y \quad U, x, V \Rightarrow z}{U, T, y \setminus x, V \Rightarrow z} \\
 /R \quad \frac{T, y \Rightarrow x}{T \Rightarrow x / y} \qquad \setminus R \quad \frac{y, T \Rightarrow x}{T \Rightarrow y \setminus x} \\
 \text{Cut} \quad \frac{T \Rightarrow x \quad U, x, V \Rightarrow y}{U, T, V \Rightarrow y}
 \end{array}$$

Lambek proved that Cut is a derived inference rule – the set of theorems is not increased by adding it to the other four inference rules. Note that this entails the decidability of L: for arbitrary sequents  $T \Rightarrow x$ , the proof procedure is guaranteed to terminate after finitely many steps with an answer to the question whether the sequent is derivable. For in each of the inference rules  $/L$ ,  $\backslash L$ ,  $/R$  and  $\backslash R$ , the number of occurrences of  $/$  and  $\backslash$  in the premises is strictly smaller than that in the conclusion (the conclusion contains a  $/$  or  $\backslash$  that is absent from the premises). Thus, establishing the derivability of the premise sequents is a more simple goal than establishing the derivability of the conclusion, and it follows that every sequent has only finitely many Cut-free derivations. One of the most important features of L as a syntactic theory is that it defines a flexible notion of constituent structure which can account for the grammaticality of sentences containing so-called non-constituent coordination:

(42) John loves and Bill hates Mary

We shall not dwell on this topic (cf., a.o., Moortgat (1988)), but focus on the semantic interpretation of L instead. That there is a strong link between syntax and semantics in L has repeatedly been stressed by Van Benthem (e.g., (1986)), who showed that the semantic interpretation of an L-derivable sequent is directly determined by the proof of its validity in the syntactic calculus: the rules  $/L$  and  $\backslash L$  correspond to functional application,  $/R$  and  $\backslash R$  to lambda abstraction, and Cut to substitution. The best-known explicit elaboration of Van Benthem's ideas can be found in Moortgat (1988). Within this semantics, every category  $x$  is associated with a type:  $\text{TYPE}(x)$ . For atomic categories, this type is stipulated, and  $\text{TYPE}(x/y) = \text{TYPE}(y/x) = (\text{TYPE}(y), \text{TYPE}(x))$ . Moreover, all categories in the left- and right-hand side of a derivable sequent  $x_1, \dots, x_n \Rightarrow x$  are assigned a semantic value, consisting of a lambda term of type  $\text{TYPE}(x_i)$  in a type-theoretic language:  $x_1 : \alpha_1, \dots, x_n : \alpha_n \Rightarrow x : \alpha$ .

In (43),  $w$  represents a variable of type  $\text{TYPE}(y)$ ;  $\alpha$ ,  $\beta$  and  $\delta$  are arbitrary lambda terms of type  $\text{TYPE}(y)$ ,  $(\text{TYPE}(y), \text{TYPE}(x))$  and  $\text{TYPE}(x)$ , respectively. Only the terms of the 'active' categories are indicated. The (invisible) terms of other categories are assumed to be identical in premise and conclusion.<sup>8</sup>

$$(43) \quad \begin{array}{l} \text{AX} \quad x : \delta \Rightarrow x : \delta \\ /L \quad \frac{T \Rightarrow y : \alpha \quad U, x : \beta(\alpha), V \Rightarrow z}{U, x/y : \beta, T, V \Rightarrow z} \end{array}$$

<sup>8</sup> The semantics is kept extensional for expository purposes. But it can be intensionalized easily. Define  $\text{TYPE}(x/y) = \text{TYPE}(y/x) = ((s, (\text{TYPE}(y))), \text{TYPE}(x))$  instead of  $(\text{TYPE}(y), \text{TYPE}(x))$ ; the terms assigned to the axioms and the Cut rule remain the same; and the semantic interpretations of  $/L$  and  $/R$  are changed in the following way ( $\backslash L$  and  $\backslash R$  are completely analogous):

$/L$ : if  $T \Rightarrow y : \alpha$  and  $U, x : \beta(\wedge \alpha), V \Rightarrow z : \gamma$ , then  $U, x/y : \beta, T, V \Rightarrow z$

$/R$ : if  $T, y : \vee w \Rightarrow x : \alpha$ , then  $T \Rightarrow x/y : \lambda w. \alpha$

$$\begin{array}{l}
\text{L} \quad \frac{T \Rightarrow y : \alpha \quad U, x : \beta(\alpha), V \Rightarrow z}{U, T, y \setminus x : \beta, V \Rightarrow z} \\
\text{/R} \quad \frac{T, y : w \Rightarrow x : \delta}{T \Rightarrow x/y : \lambda w. \delta} \quad \text{\R} \quad \frac{y : w, T \Rightarrow x : \delta}{T \Rightarrow y \setminus x : \lambda w. \delta} \\
\text{Cut} \quad \frac{T \Rightarrow x : \delta \quad U, x : \delta, V \Rightarrow y}{U, T, V \Rightarrow y}
\end{array}$$

It can be shown that if the sequent  $x_1 : \alpha_1, \dots, x_n : \alpha_n \Rightarrow x : \alpha$  is derivable in L, then there is a Cut-free L-derivation of this sequent  $x_1 : \alpha_1, \dots, x_n : \alpha_n \Rightarrow x : \beta$  with  $\alpha \leftrightarrow \beta$  (cf. Hendriks (1989)). Hence we only have to consider the (finitely many) Cut-free derivations to get all semantic interpretations assigned to a sequent.

This immediately entails that under the usual analysis, in which verbs take noun phrases (being of the atomic category np) as arguments, a sentence like (44) is unambiguous within L.

(44) Every man finds a woman

For if *every man* and *a woman* belong to the (atomic) category np, and *finds* is of category (np\s)/np, then there is only one Cut-free derivation of the sequent np, (np\s)/np, np  $\Rightarrow$  s,

$$(45) \quad \frac{\text{np} \Rightarrow \text{np} \quad \frac{\text{np} \Rightarrow \text{np} \quad \text{s} \Rightarrow \text{s}}{\text{np}, \text{np}'\text{s} \Rightarrow \text{s}}}{\text{np}, (\text{np}'\text{s})/\text{np}, \text{np} \Rightarrow \text{s}}$$

and (hence) only one interpretation for it.

However, in some cases scope ambiguities can be accounted for in L: when the following (extensional) assignment of categories, types and translations is assumed,

(46)	EXPRESSION	CATEGORY	TYPE	TRANSLATION
	<i>every man</i>	s/(n\s)	((e,t),t)	$\lambda P. \forall x[\text{MAN}(x) \rightarrow P(x)] = \alpha$
	<i>finds</i>	(n\s)/n	(e,(e,t))	$\lambda x \lambda y. \text{FIND}(x)(y) = \beta$
	<i>a woman</i>	(s/n)\s	((e,t),t)	$\lambda P. \exists y[\text{WOMAN}(y) \wedge P(y)] = \gamma$

then sentence (44) is assigned two readings:

- (47) Every man finds a woman
- (i)  $\forall x[\text{MAN}(x) \rightarrow \exists y[\text{WOMAN}(y) \wedge \text{FIND}(y)(x)]]$
  - (ii)  $\exists y[\text{WOMAN}(y) \wedge \forall x[\text{MAN}(x) \rightarrow \text{FIND}(y)(x)]]$

There are six Cut-free derivations of the sequent s/(n\s), (n\s)/n, (s/n)\s  $\Rightarrow$  s. The first one is assigned interpretation (i); the interpretation of all other derivations is equivalent to

(ii). (It is obvious that the set of Cut-free derivations still embodies a great deal of ‘spurious ambiguity’.)

$$\begin{array}{c}
 (48) \quad \frac{\frac{\frac{\frac{\frac{n \Rightarrow n}{n \Rightarrow n} \quad \frac{n \Rightarrow n}{n, n \setminus s \Rightarrow s} \quad s \Rightarrow s}{n, (n \setminus s)/n, n \Rightarrow s}}{n, (n \setminus s)/n \Rightarrow s/n} \quad s \Rightarrow s}{n, (n \setminus s)/n, (s/n) \setminus s \Rightarrow s}}{(n \setminus s)/n, (s/n) \setminus s \Rightarrow n \setminus s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \Rightarrow s} \\
 \\
 (49) \quad \frac{\frac{\frac{n \Rightarrow n}{(n \setminus s)/n, n \Rightarrow n \setminus s} \quad n \setminus s \Rightarrow n \setminus s}{s/(n \setminus s), (n \setminus s)/n, n \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n \Rightarrow s/n} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \Rightarrow s} \\
 \\
 (50) \quad \frac{\frac{\frac{\frac{n \Rightarrow n}{n \Rightarrow n} \quad \frac{n \Rightarrow n}{n, n \setminus s \Rightarrow s} \quad s \Rightarrow s}{(n \setminus s)/n, n \Rightarrow n \setminus s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, n \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n \Rightarrow s/n} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \Rightarrow s} \\
 \\
 (51) \quad \frac{\frac{\frac{\frac{n \Rightarrow n}{n \Rightarrow n} \quad \frac{n \Rightarrow n}{n, n \setminus s \Rightarrow s} \quad s \Rightarrow s}{(n \setminus s)/n, n \Rightarrow n \setminus s} \quad s \Rightarrow s}{n, (n \setminus s)/n, n \Rightarrow s} \quad s \Rightarrow s}{(n \setminus s)/n, n \Rightarrow n \setminus s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, n \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n \Rightarrow s/n} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \Rightarrow s} \\
 \\
 (52) \quad \frac{\frac{\frac{n \Rightarrow n}{n \Rightarrow n} \quad \frac{n \setminus s \Rightarrow n \setminus s}{s/(n \setminus s), n \setminus s \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, n \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n \Rightarrow s/n} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \Rightarrow s} \\
 \\
 (53) \quad \frac{\frac{\frac{\frac{n \Rightarrow n}{n \Rightarrow n} \quad \frac{s \Rightarrow s}{n, n \setminus s \Rightarrow s} \quad s \Rightarrow s}{n \setminus s \Rightarrow n \setminus s} \quad s \Rightarrow s}{n \setminus s \Rightarrow n \setminus s} \quad s \Rightarrow s}{s/(n \setminus s), n \setminus s \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, n \Rightarrow s} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n \Rightarrow s/n} \quad s \Rightarrow s}{s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \Rightarrow s}
 \end{array}$$

By way of illustration, we show the interpretation of derivations (48) and (49).

$$\begin{array}{c}
(54) \quad \frac{\frac{\frac{n \Rightarrow n}{w \Rightarrow w} \quad \frac{\frac{n \Rightarrow n}{v \Rightarrow v} \quad \frac{s \Rightarrow s}{\beta(w)(v) \Rightarrow \beta(w)(v)}}{n, n \backslash s \Rightarrow s}}{v, \beta(w) \Rightarrow \beta(w)(v)} \quad \frac{s \Rightarrow s}{\gamma(\lambda w. \beta(w)(v)) \Rightarrow \gamma(\lambda w. \beta(w)(v))}}{\frac{n, (n \backslash s)/n, n \Rightarrow s}{v, \beta, w \Rightarrow \beta(w)(v)} \quad \frac{s \Rightarrow s}{\gamma(\lambda w. \beta(w)(v)) \Rightarrow \gamma(\lambda w. \beta(w)(v))}} \\
\frac{\frac{n, (n \backslash s)/n, (s/n) \backslash s \Rightarrow s}{v, \beta, \gamma \Rightarrow \gamma(\lambda w. \beta(w)(v))} \quad \frac{s \Rightarrow s}{\delta = \alpha(\lambda v. \gamma(\lambda w. \beta(w)(v))) \Rightarrow \delta}}{\frac{(n \backslash s)/n, (s/n) \backslash s \Rightarrow n \backslash s}{\beta, \gamma \Rightarrow \lambda v. \gamma(\lambda w. \beta(w)(v))} \quad \frac{s \Rightarrow s}{\delta = \alpha(\lambda v. \gamma(\lambda w. \beta(w)(v))) \Rightarrow \delta}} \\
\frac{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s \Rightarrow s}{\alpha, \beta, \gamma \Rightarrow \alpha(\lambda v. \gamma(\lambda w. \beta(w)(v)))}
\end{array}$$

$$\begin{array}{c}
(55) \quad \frac{\frac{n \Rightarrow n}{v \Rightarrow v} \quad \frac{n \backslash s \Rightarrow n \backslash s}{\beta(v) \Rightarrow \beta(v)}}{\frac{(n \backslash s)/n, n \Rightarrow n \backslash s}{\beta, v \Rightarrow \beta(v)} \quad \frac{s \Rightarrow s}{\alpha(\beta(v)) \Rightarrow \alpha(\beta(v))}} \\
\frac{s/(n \backslash s), (n \backslash s)/n, n \Rightarrow s}{\alpha, \beta, v \Rightarrow \alpha(\beta(v))} \quad \frac{s \Rightarrow s}{\gamma(\lambda v. \alpha(\beta(v))) \Rightarrow \gamma(\lambda v. \alpha(\beta(v)))} \\
\frac{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s \Rightarrow s}{\alpha, \beta, \gamma \Rightarrow \gamma(\lambda v. \alpha(\beta(v)))}
\end{array}$$

Note that  $\alpha(\lambda v. \gamma(\lambda w. \beta(w)(v))) \Leftrightarrow$  (i), and that  $\gamma(\lambda v. \alpha(\beta(v))) \Leftrightarrow$  (ii).

However, the analysis given in (46) is a bit tricky. It is crucial that the two noun phrases belong to two different categories,  $s/(n \backslash s)$  and  $(s/n) \backslash s$ , and that these categories appear on the left- and the right-handside of the transitive verb, respectively. For none of the following sequents is L-derivable:<sup>9</sup>

$$\begin{array}{l}
(56) \quad (s/n) \backslash s, (n \backslash s)/n, s/(n \backslash s) \Rightarrow s \\
\quad \quad (s/n) \backslash s, (n \backslash s)/n, (s/n) \backslash s \Rightarrow s \\
\quad \quad s/(n \backslash s), (n \backslash s)/n, s/(n \backslash s) \Rightarrow s
\end{array}$$

An alternative account of the scope ambiguity of (44) within L assumes that verbs take noun phrases as arguments, and, moreover, that noun phrases are assigned the complex categories  $(s/n) \backslash s$  and  $s/(n \backslash s)$ . The number of interpretations now depends on which of  $s/(n \backslash s)$  or  $(s/n) \backslash s$  is chosen as the category for the subject noun phrase ( $np_1$ ) and the object noun phrase ( $np_2$ ). If  $np_1 = (s/n) \backslash s$  and  $np_2 = s/(n \backslash s)$ , only one interpretation is predicted. There are two interpretations when  $np_1 = np_2$  (it does not matter whether we take  $s/(n \backslash s)$  or  $(s/n) \backslash s$ ). The maximum is reached with  $np_1$  and  $np_2$  set equal to  $s/(n \backslash s)$  and  $(s/n) \backslash s$ , respectively, since the sequent

$$(57) \quad s/(n \backslash s), ((s/(n \backslash s)) \backslash s) / ((s/n) \backslash s), (s/n) \backslash s \Rightarrow s$$

<sup>9</sup> Precisely this prompted Lambek (1958, § 6) to propose the category  $s/(n \backslash s)$  for subject pronouns and  $(s/n) \backslash s$  for direct object pronouns, in view of *\*him likes he*.

has twenty five (!) Cut-free derivations, and five potentially different interpretations (we list only the reduced terms assigned to their conclusions):

- (58)  $\alpha(\lambda v.\beta(\gamma)(\lambda w.w(v)))$   
 (59)  $\alpha(\lambda v.\gamma(\lambda x.\beta(\lambda y.y(x)))(\lambda w.w(v)))$   
 (60)  $(\beta(\gamma))(\alpha)$   
 (61)  $\gamma(\lambda x.\alpha(\lambda v.\beta(\lambda y.y(x)))(\lambda w.w(v)))$   
 (62)  $\gamma(\lambda x.\beta(\lambda y.y(x)))(\alpha)$

When (58) through (62) are applied to the lexical translations in (63),

(63)	EXPRESSION	CATEGORY	TYPE	TRANSLATION
	<i>every man</i>	np <sub>1</sub>	((e,t),t)	$\lambda P.\forall x[\text{MAN}(x)\rightarrow P(x)] = \alpha$
	<i>finds</i>	(np <sub>1</sub> \s)/np <sub>2</sub>	(T,(T,t))	$\beta$
	<i>a woman</i>	np <sub>2</sub>	((e,t),t)	$\lambda P.\exists y[\text{WOMAN}(y)\wedge P(y)] = \gamma$ ,

where  $T = ((e,t),t)$  and  $\beta = \lambda T\lambda U.U(\lambda y.T(\lambda x.\text{FIND}(x)(y)))$ , interpretation (58), (59) and (60) collapse into reading (47)(i) (subject wide scope), and the remaining (61) and (62) represent object wide scope ((47)(ii)). ( $\beta$ , the lexical translation of *finds* in (63) encodes wide scope for the subject noun phrase, but this is not essential. With lexical wide scope for the direct object noun phrase, (47)(i) would have been expressed by (58) and (59), and (47)(ii) by (60), (61) and (62).) Interestingly, this analysis predicts three readings for *every man seeks a woman*, provided that the ‘intensional’ verb *seek* is assigned the UG-like lexical translation  $\beta = \lambda T\lambda U.U(\lambda y.\text{SEEK}(T)(y))$ :<sup>10</sup>

- (64) (i)  $\forall x[\text{MAN}(x)\rightarrow\exists y[\text{WOMAN}(y)\wedge\text{SEEK}(\lambda P.P(y))(x)]]$   
 (ii)  $\exists y[\text{WOMAN}(y)\wedge\forall x[\text{MAN}(x)\rightarrow\text{SEEK}(\lambda P.P(y))(x)]]$   
 (iii)  $\forall x[\text{MAN}(x)\rightarrow\text{SEEK}(\lambda P.\exists y[\text{WOMAN}(y)\wedge P(y)])(x)]$

(58) and (60) lead to (iii), (59) results in (i), and (61) and (62) arrive at (ii).<sup>11</sup>

It seems, then, that an adequate treatment of quantifier scope in the Lambek calculus is within reach. Still, a curious feature of our account is that the ‘internal directionality’ of noun phrases (s/(n\s) versus (s/n)\s) seems to be a non-negligible factor. Word order has an essential influence on the semantic possibilities of the calculus. When we have SVO (or OVS) order, as in (44), things turn out right. But in SOV cases like the embedded

<sup>10</sup> Since our semantics is extensional, this is not really an intensional translation. But cf. footnote 8.

<sup>11</sup> Note that the difference between (58) and (60) (or (61) and (62)) has no effect, because *seek* is only intensional in object position. In the case of doubly intensional transitive verbs the difference would matter – but in section 2 above we saw that such transitive verbs do not appear to exist. Something like *seem-to-seek* would be an example. For these hypothetical verbs, all five theoretical scope possibilities would be realized.

clause in the Dutch sentence (65) (assuming that *iedere man* =  $np_1$ , *een vrouw* =  $np_2$ , and *zoekt* =  $np_1 \backslash (np_2 \backslash s)$ , with respective translations  $\alpha$ ,  $\gamma$ , and  $\beta$ ),

- (65) Het is waar dat iedere man een vrouw zoekt  
*It is true that every man a woman seeks*  
 ‘It is true that every man seeks a woman’

we get maximally four derivations and two interpretations, (58) and (60). (That is, under optimal conditions: when *iedere man* ( $np_1$ ) and *een vrouw* ( $np_2$ ) have categories  $s/(n \backslash s)$  and  $(s/n)s^{12}$ , respectively.) (59), (61) and (62) are missing.

- (66)  $\alpha(\lambda v.(\beta(\gamma))(\lambda w.w(v)))$  = (58) [(68), (69) and (70)], and  
 (67)  $(\beta(\gamma))(\alpha)$  = (60) [(71)].

$$(68) \quad \frac{\frac{\frac{n \Rightarrow n \quad s \Rightarrow s}{n, n \backslash s \Rightarrow s}}{n \Rightarrow s/(n \backslash s)} \quad s \Rightarrow s}{n, (s/(n \backslash s)) \backslash s \Rightarrow s} \quad \frac{(s/n) \backslash s \Rightarrow (s/n) \backslash s \quad (s/(n \backslash s)) \backslash s \Rightarrow n \backslash s}{(s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow n \backslash s} \quad s \Rightarrow s}{s/(n \backslash s), (s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow s}$$

$$(69) \quad \frac{\frac{\frac{n \Rightarrow n \quad s \Rightarrow s}{n, n \backslash s \Rightarrow s}}{n \Rightarrow s/(n \backslash s)} \quad s \Rightarrow s}{n, (s/(n \backslash s)) \backslash s \Rightarrow s} \quad \frac{(s/n) \backslash s \Rightarrow (s/n) \backslash s}{n, (s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow s} \quad \frac{(s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow n \backslash s \quad s \Rightarrow s}{s/(n \backslash s), (s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow s}$$

$$(70) \quad \frac{(s/n) \backslash s \Rightarrow (s/n) \backslash s \quad \frac{s/(n \backslash s) \Rightarrow s/(n \backslash s) \quad s \Rightarrow s}{s/(n \backslash s), (s/(n \backslash s)) \backslash s \Rightarrow s}}{s/(n \backslash s), (s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow s}$$

$$(71) \quad \frac{\frac{\frac{n \Rightarrow n \quad s \Rightarrow s}{n, n \backslash s \Rightarrow s}}{n \Rightarrow s/(n \backslash s)} \quad s \Rightarrow s}{n, (s/(n \backslash s)) \backslash s \Rightarrow s} \quad \frac{(s/n) \backslash s \Rightarrow (s/n) \backslash s \quad \frac{(s/(n \backslash s)) \backslash s \Rightarrow n \backslash s \quad s \Rightarrow s}{s/(n \backslash s), (s/(n \backslash s)) \backslash s \Rightarrow s}}{s/(n \backslash s), (s/n) \backslash s, ((s/n) \backslash s) \backslash ((s/(n \backslash s)) \backslash s) \Rightarrow s}$$

This entails that with  $\beta = \lambda T \lambda U. U(\lambda y. \text{SEEK}(T)(y))$  as lexical translation of *zoekt*, only (i) and (iii) are represented. Alternatively, we could also assume the lexical (wide scope object) translation  $\lambda T \lambda U. T(\lambda x. U(\lambda y. \text{SEEK}(\lambda P. P(x))(y)))$ , but that would leave (iii)

<sup>12</sup> The same holds when both subject and object =  $s/(n \backslash s)$ . Both possibilities with subject =  $(s/n) \backslash s$  give rise to just one derivation (and reading).

unaccounted for. The third possibility, to assume the lexical translation  $\beta = \lambda T \lambda U. U(\lambda y. T(\lambda x. \text{SEEK}(T)(y)))$ , would only give (i). We are hence forced to adopt two lexical translations and to conclude that ‘it seems then, that the Lambek calculus will only be able to predict quantifier scope ambiguities in a restricted number of cases’ (Bouma (1986), p. 9).

Since it is the directionality of L which thwarts us, it is an obvious strategy to take a closer look at LP (for: Lambek calculus with Permutation, also known as Lambek-Van Benthem calculus), the non-directional version of the calculus which has been studied by Van Benthem (cf., e.g., his (1988)). (72) shows a possible formalization of LP (plus semantics). The two kinds of compound categories in L, the left- and right-searching functors  $x/y$  and  $y \backslash x$ , are collapsed into one category  $(y,x)$ , a non-directional functor from argument categories  $y$  to value categories  $x$ . (Note the direct correspondence between syntactic categories and semantic types here.) In (72), T, U, and V are finite sequences of categories, where  $T = T_1, T_2$  is non-empty. LP preserves the axioms and the Cut rule of L; it contains rules (L and R) for the introduction of the connective  $(,)$  in the left- and the right-hand side of sequents; moreover, a permutation rule, P, is added, which allows one to interchange adjacent categories (every permutation is a composition of interchanges between neighbours).

$$\begin{array}{l}
 (72) \quad \text{AX} \quad x : \delta \Rightarrow x : \delta \\
 \text{L} \quad \frac{T \Rightarrow y : \alpha \quad U, x : \beta(\alpha), V \Rightarrow z}{U, (y, x) : \beta, T, V \Rightarrow z} \\
 \text{R} \quad \frac{T_1, y : w, T_2 \Rightarrow x : \delta \quad U, y, x, V \Rightarrow z}{T \Rightarrow (y, x) : \lambda w. \delta \quad U, x, y, V \Rightarrow z} \\
 \text{P} \quad \frac{U, y, x, V \Rightarrow z}{U, x, y, V \Rightarrow z} \\
 \text{Cut} \quad \frac{T \Rightarrow x : \gamma \quad U, x : \gamma, V \Rightarrow y}{U, T, V \Rightarrow y}
 \end{array}$$

Conceived of as a syntactic theory, LP is not really interesting, because it has the following property: whenever a sequent  $T \Rightarrow x$  is LP-derivable, then for all permutations  $\pi(T)$  of T, also  $\pi(T) \Rightarrow x$  is LP-derivable. Nevertheless, it might be useful as a theory of semantic *type* change. We might, for example, restrict ourselves to sequents of categories that are derivable in our favourite syntactic theory (L, say), and let LP take care of their semantic interpretation. More formally:

$$(73) \quad \text{The compound expression } e_1, \dots, e_n \text{ is of category } x \text{ and has a translation } \alpha \text{ of type } t \text{ iff (i) for all } i, 1 \leq i \leq n, e_i \text{ is assigned category } x_i \text{ and translation } \alpha_i \text{ of type } t_i \text{ in the lexicon; (ii) } x_1, \dots, x_n \Rightarrow_L x; \text{ and (iii) } t_1 : \alpha_1, \dots, t_n : \alpha_n \Rightarrow_{LP} t : \alpha.$$

But now the permuting effects of LP will turn up in the semantics. Cf. (74) and (75). The sequent  $np, (np \backslash s) / np, np \Rightarrow s$  is L-derivable. In LP there are two non-equivalent derivations of  $e, (e, (e, t)), e \Rightarrow t$ : (76) and (77)– one too many.



(74) John loves Paris

(75) EXPRESSION	CATEGORY	TYPE	TRANSLATION
<i>John</i>	np	e	j
<i>loves</i>	(np\s)/np	(e,(e,t))	LOVE
<i>Paris</i>	np	e	p

$$(76) \frac{e:j \Rightarrow e:j \quad \frac{e:p \Rightarrow e:p \quad t:\text{LOVE}(j)(p) \Rightarrow t:\text{LOVE}(j)(p)}{(e,t):\text{LOVE}(j), e:p \Rightarrow t:\text{LOVE}(j)(p)}}{(e,(e,t)):\text{LOVE}, e:j, e:p \Rightarrow t:\text{LOVE}(j)(p)}}{e:j, (e,(e,t)):\text{LOVE}, e:p \Rightarrow t:\text{LOVE}(j)(p)}$$

$$(77) \frac{e:p \Rightarrow e:p \quad \frac{e:j \Rightarrow e:j \quad t:\text{LOVE}(p)(j) \Rightarrow t:\text{LOVE}(p)(j)}{(e,t):\text{LOVE}(p), e:j \Rightarrow t:\text{LOVE}(p)(j)}}{(e,(e,t)):\text{LOVE}, e:p, e:j \Rightarrow t:\text{LOVE}(p)(j)}}{(e,(e,t)):\text{LOVE}, e:j, e:p \Rightarrow t:\text{LOVE}(p)(j)}}{e:j, (e,(e,t)):\text{LOVE}, e:p \Rightarrow t:\text{LOVE}(p)(j)}$$

Moreover, given the assignments in (79) ( $\alpha = \lambda Q\lambda P.\forall x[Q(x)\rightarrow P(x)]$ ), sentence (78) gets two readings of type t, cf. (80) and (81).

(78) Every man walks

(79) EXPRESSION	CATEGORY	TYPE	TRANSLATION
<i>every</i>	np/cn	((e,t),((e,t),t))	$\alpha$
<i>man</i>	cn	(e,t)	MAN = $\beta$
<i>walks</i>	np\s	(e,t)	WALK = $\gamma$

$$(80) \frac{(e,t):\beta \Rightarrow (e,t):\beta \quad \frac{(e,t):\gamma \Rightarrow (e,t):\gamma \quad t:\alpha(\beta)(\gamma) \Rightarrow t:\alpha(\beta)(\gamma)}{((e,t),t):\alpha(\beta), (e,t):\gamma \Rightarrow t:\alpha(\beta)(\gamma)}}{(e,t),((e,t),t):\alpha, (e,t):\beta, (e,t):\gamma \Rightarrow t:\alpha(\beta)(\gamma) = \delta}$$

$$(81) \frac{(e,t):\gamma \Rightarrow (e,t):\gamma \quad \frac{(e,t):\beta \Rightarrow (e,t):\beta \quad t:\alpha(\gamma)(\beta) \Rightarrow t:\alpha(\gamma)(\beta)}{((e,t),t):\alpha(\gamma), (e,t):\beta \Rightarrow t:\alpha(\gamma)(\beta)}}{(e,t),((e,t),t):\alpha, (e,t):\gamma, (e,t):\beta \Rightarrow t:\alpha(\gamma)(\beta)}}{(e,t),((e,t),t):\alpha, (e,t):\beta, (e,t):\gamma \Rightarrow t:\alpha(\gamma)(\beta) = \epsilon}$$

$\delta$  is equivalent to  $\forall x[\text{MAN}(x)\rightarrow\text{WALK}(x)]$ , but  $\epsilon$  yields  $\forall x[\text{WALK}(x)\rightarrow\text{MAN}(x)]$ . Similarly, eight readings are predicted for the sentence *Every man finds a woman*. They differ as regards the scope of the quantifiers ( $\exists\forall$  versus  $\forall\exists$ ), but also with respect to the ‘ $\theta$ -roles’ of the verb and the order of the argument places of the determiner *every*. (In this particular case the symmetry of the determiner *a* avoids another multiplication of the number of readings by two.)

In the sequel we shall develop a calculus that is not thwarted by directionality (since it is a theory of semantic – i.e., non-directional – *types*), but at the same time avoids the permuting effects of LP.<sup>13</sup>

### 3.2 Flexible Montague grammar

We now give a formal definition of our flexible semantics. Each *syntactic category* C is not associated with one type, but with a(n infinite) set of types: the **type set** of C:  $T(C)$ . This set consists of a basic type,  $t_B(C)$ , together with types which are derived from that type by general rules. For the flexible version of the fragment introduced in section 1, we choose the basic types in (82).

(82)	C	$t_B(C)$
	S	t
	T	e
	CN, IV	$((s,e),t)$
	TV	$((s,e),((s,e),t))$
	PV	$((s,t),((s,e),t))$
	Det	$((s,((s,e),t)),((s,((s,e),t)),t))$

$T(C)$  is the smallest set such that:

- (83) (i)  $t_B(C) \in T(C)$ ;  
(ii) if  $(\vec{a}, b) \in T(C)$ , then  $(\vec{a}, ((s, ((s, b), t)), t)) \in T(C)$   
(cf. **value raising** = (92));  
(iii) if  $(\vec{a}, ((s, b), (\vec{c}, t))) \in T(C)$ , then  $(\vec{a}, ((s, ((s, ((s, b), t)), t)), (\vec{c}, t))) \in T(C)$   
(cf. **argument raising** = (93));  
(iv) if  $(\vec{a}, ((s, ((s, ((s, b), t)), t)), (\vec{c}, t))) \in T(C)$ , then  $(\vec{a}, ((s, b), (\vec{c}, t))) \in T(C)$   
(cf. **argument lowering** = (94)).

In this definition, s and t are specific IL types; b is an arbitrary type; and  $\vec{a}$  and  $\vec{c}$  are sequences of types:  $a_1, \dots, a_m, c_1, \dots, c_n$  with  $m$  and  $n \geq 0$ .  $(\vec{a}, b)$  represents the type  $(a_1, (\dots, (a_n, b) \dots))$ .

If we apply (ii) to  $t_B(T) = e$  ( $\vec{a}$  being empty), we see that  $TYPE(T)$  of PTQ,  $((s, ((s, e), t)), t)$ , is among the (derived) types in the type set of category T,  $T(T)$ . If we

---

<sup>13</sup> In order to be independent from specific syntactic frameworks, we will not use a sequent format for this semantic theory. We have, of course, not shown that it is altogether impossible to treat quantifier scope within flexible categorial grammar – only that current (sub-)systems (of L(P)) are inadequate (contrary to what has been suggested in Van Benthem (1986), Dowty (1988), a.o.). Thus, Moortgat (1988) simply incorporates Hendriks (1988) into L; Moortgat (1990) presents a more principled sequent-approach; and Emms (1989) offers an interesting alternative using polymorphic quantifiers and variable categories.

apply (iii) to the first (s,e) argument of  $t_B(\text{TV}) = ((s,e),((s,e),t))$  ( $\vec{a}$  again being empty), then also PTQ's  $\text{TYPE}(\text{TV}) = ((s,((s,((s,e),t)),t)),((s,e),t))$  appears to be an element of  $T(\text{TV})$ . The UG type of IVs (see above),  $((s,((s,((s,e),t)),t)),t)$ , is in  $T(\text{IV})$ : apply (ii) or (iii) to  $t_B(\text{IV}) = ((s,e),t)$ .

Similarly, every (lexical or compound) expression  $\alpha$  of category C is not assigned one translation, but a **translation set**,  $\text{Tr}(\alpha)$ : a set consisting of one or more basic translations plus derived translations:

### 3.2.1 Basic translations

#### 3.2.1.1 Basic translations of lexical expressions.

Every lexical expression  $\alpha$  is assigned a set of basic translations,  $\text{Tr}_B(\alpha)$ , in the lexicon. In general – i.e., for all lexically unambiguous expressions –  $\text{Tr}_B(\alpha)$  will only contain one member:  $\text{tr}_B(\alpha)$ . The other elements of  $\text{Tr}(\alpha)$  are derived from  $\text{tr}_B(\alpha)$  by general rules (cf. section 3.2.2, below). It is important to stress that  $\text{tr}_B(\alpha_C)$  is *not necessarily of the basic type*  $t_B(C)$ . The fragment contains the following basic translations ( $\text{Tr}_B(a) = \{\text{tr}_B(a)\}$ ):

(84)	a	$\text{tr}_B(a)_{\text{type}(\text{tr}_B(a))}$
	[CN $\text{price}$ ]	$\text{PRICE}_{((s,e),t)}$
	[CN $\text{man}$ ]	$\lambda x. [\text{MAN}(\forall x) \wedge \exists u(x=\hat{u})]_{((s,e),t)}$
	[CN $\text{woman}$ ]	$\lambda x. [\text{WOMAN}(\forall x) \wedge \exists u(x=\hat{u})]_{((s,e),t)}$
	[CN $\text{unicorn}$ ]	$\lambda x. [\text{UNICORN}(\forall x) \wedge \exists u(x=\hat{u})]_{((s,e),t)}$
	[IV $\text{be missing}$ ]	$\text{BE-MISSING}_{((s,((s,((s,e),t)),t)),t)}$
	[IV $\text{walk}$ ]	$\lambda x. \text{WALK}(\forall x)_{((s,e),t)}$
	[IV $\text{rise}$ ]	$\text{RISE}_{((s,e),t)}$
	[T $\text{John}$ ]	$j_e$
	[T $\text{Mary}$ ]	$m_e$
	[Det $a$ ]	$\lambda P \lambda Q. \exists x [\forall P(x) \wedge \forall Q(x)]_{((s,((s,e),t)),((s,((s,e),t)),t))}$
	[Det $\text{the}$ ]	$\lambda P \lambda Q. \exists x [\forall y [\forall P(y) \leftrightarrow x=y] \wedge \forall Q(x)]_{((s,((s,e),t)),((s,((s,e),t)),t))}$
	[Det $\text{every}$ ]	$\lambda P \lambda Q. \forall x [\forall P(x) \rightarrow \forall Q(x)]_{((s,((s,e),t)),((s,((s,e),t)),t))}$
	[Det $\text{one}$ ]	$\lambda P \lambda Q. \exists x \forall y [[\forall P(y) \wedge \forall Q(y)] \leftrightarrow x=y]_{((s,((s,e),t)),((s,((s,e),t)),t))}$
	[TV $\text{seek}$ ]	$\lambda T \lambda x. \text{SEEK}(T)(\forall x)_{((s,((s,((s,e),t)),t)),((s,e),t))}$
	[TV $\text{find}$ ]	$\lambda y \lambda x. \text{FIND}(\forall y)(\forall x)_{((s,e),((s,e),t))}$
	[PV $\text{claim}$ ]	$\lambda p \lambda x. \text{CLAIM}(p)(\forall x)_{((s,t),((s,e),t))}$
	[PV $\text{believe}$ ]	$\lambda p \lambda x. \text{BELIEVE}(p)(\forall x)_{((s,t),((s,e),t))}$

Note that the fragment does not contain syntactic variables  $he_j$ ; that the basic translation of *seek* is not of the basic type of transitive verb phrases,  $t_B(\text{TV}) = ((s,e),((s,e),t))$ ; that the basic translation of *be missing* is not of the basic type assigned to intransitive verb phrases,  $t_B(\text{IV}) = ((s,e),t)$ ; and that the functional application of the basic translation of *a*

to the intension of the basic translation of *woman* yields a translation of *a woman* which is not of the basic type assigned to noun phrases,  $\text{tr}_B(T) = e$ , but of the derived type  $((s,((s,e),t)),t)$ .

### 3.2.1.2 Basic translations of compound expressions

Compound expressions  $\alpha_0$  are also assigned a set of basic translations:  $\text{Tr}_B(\alpha_0)$ .  $\text{Tr}_B(\alpha_0)$  is a function of (i) the translation sets of the parts,  $\alpha_1, \dots, \alpha_n$ , of  $\alpha_0$ :  $\text{Tr}(\alpha_1), \dots, \text{Tr}(\alpha_n)$  (the sets consisting of both the basic translations and the derived translations of  $\alpha_1, \dots, \alpha_n$ ); and (ii) the semantic operation  $R$  associated with the syntactic rule which constructs  $\alpha_0$  from its immediate parts  $\alpha_1, \dots, \alpha_n$ . The fragment only contains binary rules: in all cases,  $\alpha_0$  is built up from two parts,  $\alpha$  and  $\beta$ . Hence the above can be simplified.

$$(85) \quad \text{Tr}_B(\alpha_0) = \{R(\alpha', \beta') : \alpha' \in \text{Tr}(\alpha) \text{ and } \beta' \in \text{Tr}(\beta)\}$$

There are three **semantic operations**  $R$  in the fragment: FA, GC and GD.

$$(86) \quad \mathbf{FA} \text{ ((Intensional) Functional Application): for } \gamma \text{ of type } ((s,a),b) \text{ and } \delta \text{ of type } a:$$

$$\text{FA}(\gamma, \delta) = \gamma(\wedge \delta)$$

$$(87) \quad \mathbf{GC} \text{ (Generalized Conjunction): for } \gamma \text{ and } \delta \text{ of type } (\vec{a}, t):$$

$$\text{GC}(\gamma, \delta) = \lambda \vec{x}_a. [\gamma(\vec{x}) \wedge \delta(\vec{x})]$$

$$(88) \quad \mathbf{GD} \text{ (Generalized Disjunction): for } \gamma \text{ and } \delta \text{ of type } (\vec{a}, t):$$

$$\text{GD}(\gamma, \delta) = \lambda \vec{x}_a. [\gamma(\vec{x}) \vee \delta(\vec{x})]$$

(86), (87) and (88) play a role in the **rules of the fragment**:

$$(A) \quad \text{Rules of application: } \text{Tr}_B(\alpha_0) = \{\mathbf{FA}(\alpha', \beta') : \alpha' \in \text{Tr}(\alpha) \text{ and } \beta' \in \text{Tr}(\beta)\}$$

$$(i) \quad \beta_T + \alpha_{IV} \rightarrow \alpha_0 = [{}_S \beta_T \alpha_{IV}]$$

$$(ii) \quad \alpha_{TV} + \beta_T \rightarrow \alpha_0 = [{}_{IV} \alpha_{TV} \beta_T]$$

$$(iii) \quad \alpha_{Det} + \beta_{CN} \rightarrow \alpha_0 = [{}_T \alpha_{Det} \beta_{CN}]$$

$$(iv) \quad \alpha_{PV} + \beta_S \rightarrow \alpha_0 = [{}_{IV} \alpha_{PV} \beta_S]$$

$$(B) \quad \text{Rules of conjunction: } \text{Tr}_B(\alpha_0) = \{\mathbf{GC}(\alpha', \beta') : \alpha' \in \text{Tr}(\alpha) \text{ and } \beta' \in \text{Tr}(\beta)\}$$

$$(v) \quad \alpha_S + \beta_S \rightarrow \alpha_0 = [{}_S \alpha_S \text{ and } \beta_S]$$

$$(vi) \quad \alpha_{IV} + \beta_{IV} \rightarrow \alpha_0 = [{}_{IV} \alpha_{IV} \text{ and } \beta_{IV}]$$

$$(vii) \quad \alpha_T + \beta_T \rightarrow \alpha_0 = [{}_T \alpha_T \text{ and } \beta_T]$$

$$(viii) \quad \alpha_{TV} + \beta_{TV} \rightarrow \alpha_0 = [{}_{TV} \alpha_{TV} \text{ and } \beta_{TV}]$$

- (C) Rules of disjunction:  $\text{Tr}_B(\alpha_0) = \{\mathbf{GD}(\alpha', \beta') : \alpha' \in \text{Tr}(\alpha) \text{ and } \beta' \in \text{Tr}(\beta)\}$
- (ix)  $\alpha_S + \beta_S \rightarrow \alpha_0 = [{}_S \alpha_S \text{ or } \beta_S]$
  - (x)  $\alpha_{IV} + \beta_{IV} \rightarrow \alpha_0 = [{}_{IV} \alpha_{IV} \text{ or } \beta_{IV}]$
  - (xi)  $\alpha_T + \beta_T \rightarrow \alpha_0 = [{}_T \alpha_T \text{ or } \beta_T]$
  - (xii)  $\alpha_{TV} + \beta_{TV} \rightarrow \alpha_0 = [{}_{TV} \alpha_{TV} \text{ or } \beta_{TV}]$

Note that there are no rules of quantification in the fragment. Their task will be taken over by the derived translations (see section 3.2. below). Also note that we have added rules for the conjunction and disjunction of transitive verb phrases ((viii) and (xii)), cf. sentence (29) and rule (32) above.

An example: application rule (iii) allows us to construct the noun phrase *a fish* from the determiner *a* and the common noun *fish*.

$$(89) \quad \text{tr}_B(a) = \lambda P \lambda Q. \exists y [{}^V P(y) \wedge {}^V Q(y)], \text{ and } \text{tr}_B(\textit{fish}) = \lambda x. [\text{FISH}({}^V x) \wedge \exists u (x = {}^A u)]$$

Rule (iii) states that  $\text{Tr}_B(a \textit{ fish}) = \{\mathbf{FA}(\alpha', \beta') : \alpha' \in \text{Tr}(a) \text{ and } \beta' \in \text{Tr}(\textit{fish})\}$ . Because  $\text{tr}_B(a) \in \text{Tr}(a)$  and  $\text{tr}_B(\textit{fish}) \in \text{Tr}(\textit{fish})$ , we know that (90) of type  $((s, ((s, e), t)), t)$  is a member of  $\text{Tr}_B(a \textit{ fish})$ .

$$(90) \quad \mathbf{FA}(\text{tr}_B(a), \text{tr}_B(\textit{fish})) = \lambda P \lambda Q. \exists y [{}^V P(y) \wedge {}^V Q(y)] (\wedge \lambda x. [\text{FISH}({}^V x) \wedge \exists u (x = {}^A u)]) \Leftrightarrow \lambda Q. \exists v [{}^V \text{FISH}(v) \wedge {}^V Q({}^A v)]$$

### 3.2.2 Derived translations

The **translation set**  $\text{Tr}(\alpha)$  of a (lexical or compound) expression  $\alpha$  is defined inductively:  $\text{Tr}(\alpha)$  is the smallest set such that:

- (91) For all  $\alpha' \in \text{Tr}_B(\alpha)$ :  $\alpha' \in \text{Tr}(\alpha)$ ;
- (92) **value raising (VR)**:  $(\vec{a}, b) \Rightarrow (\vec{a}, ((s, ((s, b), t)), t))$   
if  $\alpha' \in \text{Tr}(\alpha)$  and  $\alpha'$  is of type  $(\vec{a}, b)$ , then  
 $\lambda \vec{x}_a \vec{\lambda} w_{(s, ((s, b), t))}. {}^V w (\wedge [\alpha'(\vec{x})]) \in \text{Tr}(\alpha)$ ;
- (93) **argument raising (AR)**:  $(\vec{a}, ((s, b), (\vec{c}, t))) \Rightarrow (\vec{a}, ((s, ((s, ((s, b), t)), t)), (\vec{c}, t)))$   
if  $\alpha' \in \text{Tr}(\alpha)$  and  $\alpha'$  is of type  $(\vec{a}, ((s, b), (\vec{c}, t)))$ , then  
 $\lambda \vec{x}_a \vec{\lambda} w_{(s, ((s, ((s, b), t)), t))} \lambda \vec{y}_c. {}^V w (\wedge \lambda z_{(s, b)}. \alpha'(\vec{x})(z)(\vec{y})) \in \text{Tr}(\alpha)$ ; and
- (94) **argument lowering (AL)**:  $(\vec{a}, ((s, ((s, ((s, b), t)), t)), (\vec{c}, t))) \Rightarrow (\vec{a}, ((s, b), (\vec{c}, t)))$   
if  $\alpha' \in \text{Tr}(\alpha)$  and  $\alpha'$  is of type  $(\vec{a}, ((s, ((s, ((s, b), t)), t)), (\vec{c}, t)))$ , then  
 $\lambda \vec{x}_a \vec{\lambda} w_{(s, b)} \lambda \vec{y}_c. \alpha'(\vec{x}) (\wedge \lambda z_{(s, ((s, b), t))}. {}^V z(w))(\vec{y}) \in \text{Tr}(\alpha)$ .

In (92), (93) and (94)  $\vec{x}_a$  and  $\vec{y}_c$  are sequences of variables,  $\vec{x} = x_{1_{a_1}}, \dots, x_{n_{a_n}}$  and  $\vec{y} = y_{1_{c_1}}, \dots, y_{m_{c_m}}$ , of type  $\vec{a} = a_1, \dots, a_n$  and  $\vec{c} = c_1, \dots, c_m$  respectively, with  $m$  and  $n \geq 0$ .  $\lambda \vec{x}_a \tau$  abbreviates  $\lambda x_{1_{a_1}} \dots \lambda x_{n_{a_n}} \tau$ , and  $\tau(\vec{x}_a)$  abbreviates  $\tau(x_{1_{a_1}}) \dots (x_{n_{a_n}})$ . All variables in

the above terms are different.  $\lambda x_{1a_1} \dots \lambda x_{na_n} \lambda x_{n+1(b,c)} \cdot \forall x_{n+1} (\wedge [\alpha'(x_{1a_1}) \dots (x_{na_n})])$  is, for example, the term assigned by value raising.

First some lexical examples. When value raising is applied to the basic translation ( $\text{tr}_B$ ) of *John*,  $j$ , we get  $\lambda P. \forall P(\wedge j)$  as a derived translation, i.e., the translation PTQ assigns to this noun phrase; application of argument raising to (the first argument of) the basic translation of *find*,  $\lambda y \lambda x. \text{FIND}(\forall y)(\forall x)$ , leads to  $\lambda T \lambda x. \forall T(\wedge \lambda y. \text{FIND}(\forall y)(\forall x))$ , the PTQ translation; application of argument lowering to (the first argument of) the basic translation of *seek*,  $\lambda T \lambda x. \text{SEEK}(T)(\forall x)$ , gives us a derived translation that can be applied to the intension of the basic translations of proper names (which are of type  $e$ ):  $\lambda y \lambda z. \text{SEEK}(\wedge \lambda P. \forall P(\wedge y))(\forall z)$ .

It is worth noting that the semantic operations FA, GC and GD are *partial* with respect to  $\text{Tr}(\alpha)$  and  $\text{Tr}(\beta)$ . Sometimes,  $\text{FA/GC/GD}(\alpha', \beta')$  fails to be defined for  $\alpha' \in \text{Tr}(\alpha)$  and  $\beta' \in \text{Tr}(\beta)$ . For instance,  $\text{Tr}_B(\textit{John and Mary})$  does not contain the GC of  $j_e$  and  $m_e$ , simply because  $j$  and  $m$  are not of a conjoinable type  $(\vec{a}, t)$ . Neither does  $\text{Tr}_B(\textit{seek John})$  contain the FA of  $\lambda T \lambda x. \text{SEEK}(T)(\forall x)$  – an expression of type  $((s, ((s, ((s, e), t)), t)), ((s, e), t))$  – and  $j_e$ , for the reason that  $\text{FA}(\lambda T \lambda x. \text{SEEK}(T)(\forall x), j)$  is undefined:  $\lambda T \lambda x. \text{SEEK}(T)(\forall x)$  is not of the required type  $((s, \text{TYPE}(j)), b) = ((s, e), b)$ . But this is no problem. Since we also have the derived translations  $\lambda P. \forall P(\wedge j)$  and  $\lambda P. \forall P(\wedge m)$  at our disposal, we can be sure that  $\text{Tr}_B(\textit{John and Mary})$  and  $\text{Tr}_B(\textit{seek John})$  are non-empty:

- (95)  $\text{GC}(\lambda P. \forall P(\wedge j), \lambda P. \forall P(\wedge m)) =$   
 $\lambda Q. [\lambda P. \forall P(\wedge j)(Q) \wedge \lambda P. \forall P(\wedge m)(Q)] \in \text{Tr}_B(\textit{John and Mary})$ , and  
 (96)  $\text{FA}(\lambda T \lambda x. \text{SEEK}(T)(\forall x), \lambda P. \forall P(\wedge j)) =$   
 $\lambda T \lambda x. \text{SEEK}(T)(\forall x)(\wedge \lambda P. \forall P(\wedge j)) \in \text{Tr}_B(\textit{seek John})$ .

Let us now consider a compound example, which shows that we are able to account for the *de dicto* wide scope-*or* reading (24) of sentence (19): rule (iii) enables us to construct *a fish* from the determiner *a* and the common noun *fish*. We have just seen that

- (97)  $\text{FA}(\text{tr}_B(a), \text{tr}_B(\textit{fish})) = \lambda P \lambda Q. \exists y [\forall P(y) \wedge \forall Q(y)] (\wedge \lambda x. [\text{FISH}(\forall x) \wedge \exists u (x = \wedge u)]) \Leftrightarrow$   
 $\lambda Q. \exists v [\forall \text{FISH}(v) \wedge \forall Q(\wedge v)] (= (90))$

of type  $((s, ((s, ((s, e), t)), t)), t) = \pi$  ( $\pi$  is Montague's  $\text{TYPE}(T)$ ) is a member of  $\text{Tr}_B(\textit{a fish})$ . Hence, according to (91), it is also a member  $\text{Tr}(\textit{a fish})$ . Due to (92), value raising, we know that  $\text{Tr}(\textit{a fish})$  furthermore contains the derived translation (98), of type  $((s, ((s, \pi), t)), t) = ((s, ((s, ((s, ((s, e), t)), t)), t)), t) = \omega$ :

- (98)  $\lambda Q_{(s, ((s, ((s, ((s, e), t)), t)), t))}. \forall Q(\wedge \lambda P. \exists v [\text{FISH}(v) \wedge \forall P(\wedge v)])$

Analogously, we can show that the noun phrase *a bike* has the translation

- (99)  $\lambda Q_{(s, ((s, ((s, ((s, e), t)), t)), t))}. \forall Q(\wedge \lambda P. \exists v [\text{BIKE}(v) \wedge \forall P(\wedge v)])$ .

Therefore, the rule of disjunction (x) specifies  $GD((98),(99)) = (100)$  as one of the basic translations of *a fish or a bike*:

$$(100) \quad \lambda I. [\lambda Q. \forall Q (\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)])(I) \vee \lambda Q. \forall Q (\wedge \lambda P. \exists v [BIKE(v) \wedge \forall P(\wedge v)])(I)] \\ \Leftrightarrow \lambda I. [\forall I (\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)]) \vee \forall I (\wedge \lambda P. \exists v [BIKE(v) \wedge \forall P(\wedge v)]]]$$

(100) is also of type  $\omega$ . Note that we can use argument raising (applying it to the first argument) to blow up the basic translation of *seek*,  $\lambda T \lambda x. SEEK(T)(\forall x)$ , which is of type  $((s, ((s, ((s, e), t)), (s, e), t)), (s, e), t)$ , to (101), which is of the higher type  $((s, \omega), ((s, e), t)) = ((s, ((s, ((s, ((s, ((s, e), t)), t)), t)), t)), (s, e), t)$ :

$$(101) \quad \lambda J_{(s, \omega)} \lambda y. \forall J (\wedge \lambda T. [\lambda T' \lambda x. SEEK(T')(\forall x)](T)(y)) \Leftrightarrow \lambda J \lambda y. \forall J (\wedge \lambda T. SEEK(T)(\forall y))$$

Because  $(100) \in Tr(a \text{ fish or a bike})$  and  $(101) \in Tr(seek)$ , we can apply rule (ii), which gives us  $FA((101), (100)) = (102)$  – of type  $((s, e), t)$  – as one of the basic translations of *seek a fish or a bike*:

$$(102) \quad [\lambda J \lambda y. \forall J (\wedge \lambda T. SEEK(T)(\forall y))] \\ (\wedge \lambda I. [\forall I (\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)]) \vee \forall I (\wedge \lambda P. \exists v [BIKE(v) \wedge \forall P(\wedge v)]]]) \Leftrightarrow \\ \lambda y. [SEEK(\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)])(\forall y) \vee SEEK(\wedge \lambda P. \exists v [BIKE(v) \wedge \forall P(\wedge v)])(\forall y)]$$

Finally, when (102) is applied to  $\wedge j$  (rule (i)), we arrive at reading (24):

$$(103) \quad [\lambda y. [SEEK(\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)])(\forall y) \vee \\ SEEK(\wedge \lambda P. \exists v [BIKE(v) \wedge \forall P(\wedge v)])(\forall y)]] (\wedge j) \Leftrightarrow \\ SEEK(\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)])(j) \vee SEEK(\wedge \lambda P. \exists v [FISH(v) \wedge \forall P(\wedge v)])(j)$$

#### 4. APPLICATIONS

This section shows a number of applications of the flexible Montague grammar defined above. In section 4.1 we shall see that we can represent ambiguities due to the occurrence of quantified noun phrases, not only in ‘flat’ structures (where the quantifiers are all immediate arguments of one relation (4.1.1)), but also in embedded application structures (4.1.2) of arbitrary complexity. In 4.2 we shall consider ambiguities regarding the scope of coordination. In 4.3 it will be shown that certain restrictions on possible scopings can be readily accounted for within the present framework. Let us first settle some notation conventions.

(104) An application of VR (AR, AL) on a type  $(\vec{a}, c)$  will be called an **m-application** of VR (AR, AL) (written as  ${}^m VR ({}^m AR, {}^m AL)$ ) iff  $length(\vec{a}) = m-1$ . [ $m \in \{1, 2, 3, \dots\}$ , cf. (92), (93) and (94) for  $\vec{a}$ .]

Example:  $(a,b) \Rightarrow^2 \text{VR } (a,((s,((s,b),t)),t))$ ;  $(a,b) \Rightarrow^1 \text{VR } ((s,((s,(a,b)),t)),t)$ .

(105) For any type  $a$ :  $a^0 = a$ ;  $a^{n+1} = ((s,a^n),t)$ .

Example:  $e^2 = ((s,((s,e),t)),t)$ .

(106) A **reading** of an expression  $\alpha$  is an interpretation of  $\alpha$  of type  $t$ .

#### 4.1 The scope of quantification

##### 4.1.1 Flat structures

Sentence (107) consists of an extensional transitive verb and two proper names.

(107) John finds Mary

If we apply rule (ii), and use the basic translations of *find* and *Mary*, we obtain

(108)  $\lambda y \lambda x. \text{FIND}(\forall y)(\forall x)(\wedge m) \Leftrightarrow \lambda x. \text{FIND}(\forall \wedge m)(\forall x) \Leftrightarrow \lambda x. \text{FIND}(m)(\forall x)$ .

If we then apply (108) to the intension of the basic translation  $j$  of *John* (with the help of rule (i)), the resulting translation is

(109)  $\lambda x. \text{FIND}(m)(\forall x)(\wedge j) \Leftrightarrow \text{FIND}(m)(j)$ .

It can be shown that this is the only reading (in the sense of (106)) that is assigned to sentence (107) by the flexible grammar. I.e., if we, instead of using the basic translations, were to combine arbitrary derived translations, the result would be equivalent (the proofs of claims about numbers of readings can be found in or deduced from the Appendix).

If an intensional transitive verb is combined with two proper names, the grammar also assigns one reading, regardless of the translations used:

(110) John seeks Mary

We shall sketch two possible ways to obtain this reading. One way involves the application of the basic translation of *seek*,  $\lambda T \lambda x. \text{SEEK}(T)(\forall x)$ , to the intension of the derived translation  $\lambda P. \forall P(\wedge m)$  of *Mary* (the result of value-raising  $m$ ) and the combination of the result with  $j$ , just as we did in the case of (107). But we can also apply argument lowering to the first argument of  $\lambda T \lambda x. \text{SEEK}(T)(\forall x)$ ,

(111)  $\lambda y. \lambda T \lambda x. \text{SEEK}(T)(\forall x)(\wedge \lambda P. \forall P(y)) \Leftrightarrow \lambda y \lambda x. \text{SEEK}(\wedge \lambda P. \forall P(y))(\forall x)$ ,



and combine the result (via (ii) and (i)) with the basic translations of the proper names:  
 $[\lambda y \lambda x. \text{SEEK}(\lambda P. \forall P(y))(\forall x)](\wedge m)(\wedge j) \Leftrightarrow \text{SEEK}(\lambda P. \forall P(\wedge m))(j)$ .

If, however, the noun phrases surrounding the transitive verb are quantified noun phrases, then the grammar assigns a different number of readings to the sentences with *seek* and *find*:

- (112) Every man finds a woman  
 (i)  $\forall v[\text{MAN}(v) \rightarrow \exists u[\text{WOMAN}(u) \wedge \text{FIND}(u)(v)]]$   
 (ii)  $\exists u[\text{WOMAN}(u) \wedge \forall v[\text{MAN}(v) \rightarrow \text{FIND}(u)(v)]]$
- (113) Every man seeks a woman  
 (i)  $\forall v[\text{MAN}(v) \rightarrow \exists u[\text{WOMAN}(u) \wedge \text{SEEK}(\lambda P. \forall P(\wedge u))(v)]]$   
 (ii)  $\exists u[\text{WOMAN}(u) \wedge \forall v[\text{MAN}(v) \rightarrow \text{SEEK}(\lambda P. \forall P(\wedge u))(v)]]$   
 (iii)  $\forall v[\text{MAN}(v) \rightarrow \text{SEEK}(\lambda P. \exists u[\text{WOMAN}(u) \wedge \forall P(\wedge u)])(v)]$

(112) gets two readings, whereas (113) can be understood in three ways. The basic translation  $\lambda y \lambda x. \text{FIND}(\forall y)(\forall x)$  is of type  $((s,e),((s,e),t))$ , and there are two non-equivalent ways, (114) and (115), to argument-raise it in two steps to type  $((s,((s,((s,e),t),t)),((s,((s,((s,e),t),t),t)),t)) = ((s,e^2),((s,e^2),t))$ :<sup>14</sup>

- (114)  $((s,e),((s,e),t)) \Rightarrow ((s,e^2),((s,e),t)): \lambda T_1 \lambda y'. \forall T_1'(\wedge \lambda x. \text{FIND}(\forall x)(\forall y')) = \alpha$ ;  
 $((s,e^2),((s,e),t)) \Rightarrow ((s,e^2),((s,e^2),t)): \lambda T_1 \lambda T_2. \forall T_2(\wedge \lambda y. \alpha(T_1)(y)) \Leftrightarrow$   
 $\lambda T_1 \lambda T_2. \forall T_2(\wedge \lambda y. \forall T_1(\wedge \lambda x. \text{FIND}(\forall x)(\forall y)))$
- (115)  $((s,e),((s,e),t)) \Rightarrow ((s,e),((s,e^2),t)): \lambda x' \lambda T_2'. \forall T_2'(\wedge \lambda y. \text{FIND}(\forall x')(\forall y)) = \beta$ ;  
 $((s,e),((s,e^2),t)) \Rightarrow ((s,e^2),((s,e^2),t)): \lambda T_1 \lambda T_2. \forall T_1(\wedge \lambda x. \beta(x)(T_2)) \Leftrightarrow$   
 $\lambda T_1 \lambda T_2. \forall T_1(\wedge \lambda x. \forall T_2(\wedge \lambda y. \text{FIND}(\forall x)(\forall y)))$

The combination of (114) and the basic translations *every man* and *a woman* yields (112) (i); (112) (ii) is the result of combining (115) with these basic translations. Again, additional type change does not increase the number of readings. In the case of (113) we can let the argument-lowered translation of *seek*,  $\lambda y \lambda x. \text{SEEK}(\lambda P. \forall P(y))(\forall x)$ , undergo (a) and (b) as well. We then get (113) (i) and (113) (ii), respectively. But there is also a third possibility here. We can use the basic translation of *seek*,  $\lambda T \lambda x. \text{SEEK}(T)(\forall x)$ , and raise its second argument to type  $(s,e^2)$ :  $\lambda T_1 \lambda T_2. \forall T_2(\wedge \lambda y. \text{SEEK}(T_1)(\forall y))$ . In this way we account for the *de dicto* reading, (113) (iii).

In general, the combination of an n-ary relation of type  $((s,e)_1, (\dots ((s,e)_n, t)))$  with n quantified noun phrases – i.e., noun phrases of type  $((s, ((s,e), t)), t)$  – yields maximally  $n! = n \cdot (n-1) \cdot \dots \cdot 2 \cdot 1$  readings of type t (n! is a maximum here, because quantified noun phrases that are value-raised terms of type e (proper names), for instance do not help multiplying the number of readings), which is what we need.

<sup>14</sup> Different from AL, the order in which VR and AR are applied is important.

## 4.1.2 Embedded structures

The flexible grammar of section 3.2 is able to represent scope ambiguities in arbitrarily complex structures of functional application. Consider the syntactic structure of the compound sentence *Fred claims that every schoolboy believes that a mathematician wrote Through the Looking Glass*.

- (116) [<sub>S</sub> Fred<sub>T</sub> [<sub>IV</sub> claims that<sub>PV</sub> [<sub>S</sub> [<sub>T</sub> every<sub>Det</sub> schoolboy<sub>CN</sub> ] [<sub>IV</sub> believes that<sub>PV</sub> [<sub>S</sub> [<sub>TaDet</sub> mathematician<sub>CN</sub>] [<sub>IV</sub> wrote<sub>TV</sub> *Through the looking glass*<sub>T</sub> ]]]]]]]

With the basic translations indicated below, (116) gets exactly the eight readings which result from interchanging the mutual scopings of the quantified noun phrases *every schoolboy* and *a mathematician* on the one hand, and interpreting these noun phrases more or less *de dicto* or *de re* within the compound intensional context (...*believe that*(...*claim that* (...))) on the other:

<i>Fred</i>	$f_e$ ,
<i>believe that</i>	$\lambda p \lambda x. \text{BELIEVE}(\forall x, p)_{((s,t),((s,e),t))}$ ,
<i>every schoolboy</i>	$\lambda P. \forall v[\text{BOY}(v) \rightarrow \forall P(\wedge v)]_{((s,((s,e),t)),t)}$ ,
<i>claim that</i>	$\lambda p \lambda x. \text{CLAIM}(\forall x, p)_{((s,t),((s,e),t))}$ ,
<i>a mathematician</i>	$\lambda P. \exists u[\text{MATH}(u) \wedge \forall P(\wedge u)]_{((s,((s,e),t)),t)}$ ,
<i>write</i>	$\lambda y \lambda x. \text{WRITE}(\forall x, \forall y)_{((s,e),((s,e),t))}$ ,
Through the looking glass	$L_e$ .

- (117)  $\text{CLAIM}(f, \wedge \forall v[\text{BOY}(v) \rightarrow \text{BELIEVE}(v, \wedge \exists u[\text{MATH}(u) \wedge \text{WRITE}(u, L)])])$   
(118)  $\text{CLAIM}(f, \wedge \forall v[\text{BOY}(v) \rightarrow \exists u[\text{MATH}(u) \wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])$   
(119)  $\text{CLAIM}(f, \wedge \exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])$   
(120)  $\exists u[\text{MATH}(u) \wedge \text{CLAIM}(f, \wedge \forall v[\text{BOY}(v) \rightarrow \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])$   
(121)  $\forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, \wedge \text{BELIEVE}(v, \wedge \exists u[\text{MATH}(u) \wedge \text{WRITE}(u, L)])])$   
(122)  $\forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, \wedge \exists u[\text{MATH}(u) \wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])$   
(123)  $\forall v[\text{BOY}(v) \rightarrow \exists u[\text{MATH}(u) \wedge \text{CLAIM}(f, \wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])$   
(124)  $\exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, \wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])$

By way of illustration, we show the derivation of (124), semantically the most ‘dislocated’ reading of sentence (116).<sup>15</sup>

The noun phrase *a mathematician* must ‘escape from’ its original scope domain, the one embedded in *believe that*. This can be effectuated by subsequently applying <sup>3</sup>VR (creating the ‘escape hatch’) and <sup>2</sup>AR (letting the subject argument ‘go out’ of it) to the basic translation of *write*, the main functor in that domain:  $((s,e),((s,e),t)) \Rightarrow ((s,e),((s,e),t^2)) \Rightarrow ((s,e),((s,e^2),t^2))$ :

$$(125) \quad \lambda y \lambda x. \text{WRITE}(\forall x, \forall y) \Rightarrow^3 \text{VR} \lambda y \lambda x \lambda w. \forall w (\wedge \text{WRITE}(\forall x, \forall y)) \Rightarrow^2 \text{AR} \\ \lambda y \lambda T \lambda w. \forall T (\wedge \lambda x. \forall w (\wedge \text{WRITE}(\forall x, \forall y)))$$

The noun phrase *every schoolboy* is to have wider scope than *claim that*. The main functor of the domain embedded in *claim that* is *believe that*. We can perform the same trick here: first apply <sup>3</sup>VR, and then <sup>2</sup>AR, in this case to *believe that*. Due to our application of value raising to *write*, the compound first argument of *believe that* (*a mathematician wrote* Through the Looking Glass) is too large (being of type  $t^2$  instead of  $t$ ). This calls for an additional application of AR to *believe that*, <sup>1</sup>AR this time. Since *a mathematician* is going to have wider scope than *every schoolboy*, <sup>1</sup>AR should be applied after <sup>3</sup>VR and <sup>2</sup>AR.

$$(126) \quad \lambda p \lambda x. \text{BELIEVE}(\forall x, p) \Rightarrow^3 \text{VR} \lambda p \lambda x \lambda w. \forall w (\wedge \text{BELIEVE}(\forall x, p)) \Rightarrow^2 \text{AR} \\ \lambda p \lambda T \lambda w. \forall T (\wedge \lambda x. \forall w (\wedge \text{BELIEVE}(\forall x, p))) \Rightarrow^1 \text{AR} \\ \lambda Q \lambda T \lambda w. \forall Q (\wedge \lambda p. \forall T (\wedge \lambda x. \forall w (\wedge \text{BELIEVE}(\forall x, p))))$$

<sup>15</sup> The remaining seven readings can be obtained in the following way:

- |       |                 |   |
|-------|-----------------|---|
| (127) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^2 \text{AR} ((s,t),((s,e^2),t))$   |
| (128) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^3 \text{VR} ((s,e),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t^2))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t)) \Rightarrow^2 \text{AR} ((s,t^2),((s,e^2),t))$   |
| (129) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^3 \text{VR} ((s,e),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t^2))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^2 \text{AR} ((s,t),((s,e^2),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e^2),t))$   |
| (130) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^3 \text{VR} ((s,e),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t^2))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^2 \text{AR} ((s,t),((s,e^2),t)) \Rightarrow^3 \text{VR} ((s,t),((s,e^2),t^2)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e^2),t^2))$ |
|       | <i>claim:</i>   | $((s,t),((s,e),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t))$   |
| (131) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^3 \text{VR} ((s,t),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,t),((s,e^2),t^2))$   |
|       | <i>claim:</i>   | $((s,t),((s,e),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t))$   |
| (132) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^3 \text{VR} ((s,e),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t^2))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t)) \Rightarrow^3 \text{VR} ((s,t^2),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,t^2),((s,e^2),t^2))$ |
|       | <i>claim:</i>   | $((s,t),((s,e),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t))$   |
| (133) | <i>write:</i>   | $((s,e),((s,e),t)) \Rightarrow^3 \text{VR} ((s,e),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,e),((s,e^2),t^2))$   |
|       | <i>believe:</i> | $((s,t),((s,e),t)) \Rightarrow^3 \text{VR} ((s,t),((s,e),t^2)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t^2)) \Rightarrow^2 \text{AR} ((s,t^2),((s,e^2),t^2))$ |
|       | <i>claim:</i>   | $((s,t),((s,e),t)) \Rightarrow^1 \text{AR} ((s,t^2),((s,e),t))$   |

As a consequence of our applying value raising to *believe that*, the compound first argument of *claim that (every schoolboy believes that a mathematician wrote Through the Looking Glass)* does not ‘fit’ any more. This necessitates an application of <sup>1</sup>AR to *claim that*.

$$(127) \quad \lambda p \lambda x. \text{CLAIM}(\forall x, p) \Rightarrow {}^1\text{AR} \lambda Q \lambda x. \forall Q(\wedge \lambda p. \text{CLAIM}(\forall x, p))$$

With this amount of type change, we get the following translation for *a mathematician wrote Through the Looking Glass*:

$$(128) \quad [\lambda y \lambda T \lambda w. \forall T(\wedge \lambda x. \forall w(\wedge \text{WRITE}(\forall x, \forall y)))] (\wedge L) (\wedge \lambda P. \exists u[\text{MATH}(u) \wedge \forall P(\wedge u)] \Leftrightarrow \lambda w. \exists u[\text{MATH}(u) \wedge \forall w(\wedge \text{WRITE}(u, L))]$$

*Every schoolboy believes that a mathematician wrote Through the Looking Glass* is translated as:

$$(129) \quad [\lambda Q \lambda T \lambda w. \forall Q(\wedge \lambda p. \forall T(\wedge \lambda x. \forall w(\wedge \text{BELIEVE}(\forall x, p))))] (\wedge \lambda w'. \exists u[\text{MATH}(u) \wedge \forall w'(\wedge \text{WRITE}(u, L))]) (\wedge \lambda P. \forall v[\text{BOY}(v) \rightarrow \forall P(\wedge v)] \Leftrightarrow \lambda w. \exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \forall w(\wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])]$$

And we get the following translation for sentence (116) as a whole:

$$(130) \quad [\lambda Q \lambda x. \forall Q(\wedge \lambda p. \text{CLAIM}(\forall x, p))] (\wedge \lambda w. \exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \forall w(\wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])] (\wedge f) \Leftrightarrow \exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, \wedge \text{BELIEVE}(v, \wedge \text{WRITE}(u, L))]])]$$

#### 4.2 The scope of coordination

We showed in section 3.2.2 that the *de dicto* wide scope-or reading (iii) of sentence (131) can be represented.

(131) John seeks a fish or a bike

- (i)  $\text{SEEK}(\wedge \lambda P. [\exists v[\text{FISH}(v) \wedge \forall P(\wedge v)] \vee \exists v[\text{BIKE}(v) \wedge \forall P(\wedge v)]])(j)$
- (ii)  $\exists v[\text{FISH}(v) \wedge \text{SEEK}(\wedge \lambda P. \forall P(\wedge v))](j) \vee \exists v[\text{BIKE}(v) \wedge \text{SEEK}(\wedge \lambda P. \forall P(\wedge v))](j)$
- (iii)  $\text{SEEK}(\wedge \lambda P. \exists v[\text{FISH}(v) \wedge \forall P(\wedge v)])(j) \vee \text{SEEK}(\wedge \lambda P. \exists v[\text{BIKE}(v) \wedge \forall P(\wedge v)])(j)$

It is obvious that the ‘standard’ *de dicto* reading (i) is accounted for too: for this reading, it is sufficient to combine the basic translations of the expressions. The *de re* reading (ii) can be realised by building up (132),

$$(132) \quad \lambda P. [\exists v[\text{FISH}(v) \wedge \forall P(\wedge v)] \vee \exists v[\text{BIKE}(v) \wedge \forall P(\wedge v)]],$$

the generalized disjunction of the basic translations of *a fish* and *a bike*, and applying (133) to the intension of this expression:

$$(133) \quad \lambda T \lambda x. \forall T (\wedge \lambda y. \text{SEEK}(\wedge \lambda P. \forall P(y))(\forall x))$$

(133) is a derived translation of *seek* which results from applying <sup>1</sup>AL and <sup>1</sup>AR (in that order) to the basic translation:  $((s, e^2), ((s, e), t)) \Rightarrow ((s, e), ((s, e), t)) \Rightarrow ((s, e^2), ((s, e), t))$ .

$$(134) \quad \lambda T \lambda x. \text{SEEK}(T)(\forall x) \Rightarrow^1 \text{AL} \lambda y \lambda x. \text{SEEK}(\wedge \lambda P. \forall P(y))(\forall x) \Rightarrow^1 \text{AR} \\ \lambda T \lambda x. \forall T (\wedge \lambda y. \text{SEEK}(\wedge \lambda P. \forall P(y))(\forall x))$$

Also the scope ambiguity in sentence (135) is represented:

(135) John caught and ate a fish

$$(i) \quad \exists v [\text{FISH}(v) \wedge \text{CATCH}(v)(j)] \wedge \exists v [\text{FISH}(v) \wedge \text{EAT}(v)(j)]$$

$$(ii) \quad \exists v [\text{FISH}(v) \wedge \text{CATCH}(v)(j) \wedge \text{EAT}(v)(j)]$$

The extensional transitive verbs *caught* and *ate* have basic translations of the minimal type,  $t_B(\text{TV}) = ((s, e), ((s, e), t))$ , and translations of type  $((s, e^2), ((s, e), t))$  are derivable from the basic translations by raising the first argument of  $t_B(\text{TV})$ :

$$(136) \quad \lambda T \lambda x. \forall T (\wedge \lambda y. \text{CATCH}(\forall y)(\forall x)) \text{ and } \lambda T \lambda x. \forall T (\wedge \lambda y. \text{EAT}(\forall y)(\forall x)).$$

Reading (i) employs the generalized conjunction of these derived translations:

$$(137) \quad \lambda T \lambda x. [\forall T (\wedge \lambda y. \text{CATCH}(\forall y)(\forall x)) \wedge \forall T (\wedge \lambda y. \text{EAT}(\forall y)(\forall x))]$$

Reading (ii) is achieved via generalized conjunction of the basic translations, (138), followed by raising the first argument of the conjunction: (139).

$$(138) \quad \lambda y \lambda x. [\text{CATCH}(\forall y)(\forall x) \wedge \text{EAT}(\forall y)(\forall x)]$$

$$(139) \quad \lambda T \lambda x. \forall T (\wedge \lambda y. [\text{CATCH}(\forall y)(\forall x) \wedge \text{EAT}(\forall y)(\forall x)])$$

Partee and Rooth (1983) in fact only accept reading (ii), ‘unless the sentence [is] given a very marked intonation or the context is heavily loaded’ (p. 365), but this observation is at least questionable, in consideration of the natural continuation in (140):

(140) John caught and ate a fish. The fish he caught was inedible, and the fish he ate caught his eye.

Be this as it may, less probable but possible readings should be predicted by the grammar as well, preferably together with an explanation of their lesser probability. An explanation of cases like these will be given below.

The flexible grammar also supplies a desired *de dicto* wide scope-or reading of sentence (141), viz., (142), a problematic case for which Partee and Rooth, surprisingly, suggest ‘a rule quantifying in terms of a higher type’ (p. 376):

- (141) John believes that a man or a woman walks  
 (142) BELIEVE( $j, \wedge \exists u[\text{MAN}(u) \wedge \text{WALK}(u)] \vee \text{BELIEVE}(j, \wedge \exists u[\text{WOMAN}(u) \wedge \text{WALK}(u)]$ )

To get (142), we value-raise the noun phrases *a man* and *a woman*:  $((s, ((s, e), t)), t) \Rightarrow^1 \text{VR } ((s, ((s, e), t)), t)^2 = e^4$ , and take the generalized disjunction of the derived translations (I is of type  $((s, ((s, e), t)), t)^1 = e^3$ ):

- (143)  $\lambda I. [\vee I(\wedge \lambda P. \exists u[\text{MAN}(u) \wedge \vee P(\wedge u)]) \vee \vee I(\wedge \lambda P. \exists u[\text{WOMAN}(u) \wedge \vee P(\wedge u)])]$

Next we apply  $^1\text{AR}$ ,  $^2\text{VR}$ , and  $^1\text{AR}$  to  $\lambda x. \text{WALK}(\vee x)$ , the basic translation of *walk*.  $((s, e), t) \Rightarrow^1 \text{AR } ((s, e^2), t) \Rightarrow^2 \text{VR } ((s, e^2), t^2) \Rightarrow^1 \text{AR } ((s, e^4), t^2)$ :

- (144)  $\lambda x. \text{WALK}(\vee x) \Rightarrow^1 \text{AR } \lambda T. \vee T(\wedge \lambda x. \text{WALK}(\vee x)) \Rightarrow^2 \text{VR}$   
 $\lambda T \lambda w. \vee w(\vee T(\wedge \lambda x. \text{WALK}(\vee x))) \Rightarrow^1 \text{AR } \lambda Q \lambda w. \vee Q(\wedge \lambda T. \vee w(\vee T(\wedge \lambda x. \text{WALK}(\vee x))))$

(144) is applied to the intension of (143):

- (145)  $[\lambda Q \lambda t. \vee Q(\wedge \lambda T. \vee t(\vee T(\wedge \lambda x. \text{WALK}(\vee x))))]$   
 $(\wedge \lambda I. [\vee I(\wedge \lambda P. \exists u[\text{MAN}(u) \wedge \vee P(\wedge u)]) \vee \vee I(\wedge \lambda P. \exists u[\text{WOMAN}(u) \wedge \vee P(\wedge u)])]) \Leftrightarrow$   
 $\lambda t. [\vee t(\wedge \exists u[\text{MAN}(u) \wedge \text{WALK}(u)]) \vee \vee t(\wedge \exists u[\text{WOMAN}(u) \wedge \text{WALK}(u)])]$

Finally, the first argument of *believe that* is raised, and the result (146) is subsequently applied to the intension of (145) and  $j$ :

- (146)  $\lambda p \lambda x. \text{BELIEVE}(p)(\vee x) \Rightarrow^1 \text{AR } \lambda R \lambda x. \vee R(\wedge \lambda p. \text{BELIEVE}(p)(\vee x))$   
 (147)  $[\lambda R \lambda x. \vee R(\wedge \lambda p. \text{BELIEVE}(p)(\vee x))]$   
 $(\wedge \lambda t. [\vee t(\wedge \exists u[\text{MAN}(u) \wedge \text{WALK}(u)]) \vee \vee t(\wedge \exists u[\text{WOMAN}(u) \wedge \text{WALK}(u)])]) (\wedge j) \Leftrightarrow$   
 $\text{BELIEVE}(\wedge \exists u[\text{MAN}(u) \wedge \text{WALK}(u)])(j) \vee \text{BELIEVE}(\wedge \exists u[\text{WOMAN}(u) \wedge \text{WALK}(u)])(j)$

The flexible grammar of section 3 assigns all possible coordination scopes.<sup>16</sup> For example, (148) is assigned the readings (i), (ii) and (iii):

- (148) John said that every student lost or won  
 (i) SAY( $j, \wedge \forall v[\text{STUDENT}(v) \rightarrow [\text{LOOSE}(v) \vee \text{WIN}(v)]]$ )  
 (ii) SAY( $j, \wedge [\forall v[\text{STUDENT}(v) \rightarrow \text{LOOSE}(v)]] \vee \forall v[\text{STUDENT}(v) \rightarrow \text{WIN}(v)]]$ )  
 (iii) SAY( $j, \wedge \forall v[\text{STUDENT}(v) \rightarrow \text{LOOSE}(v)]$ )  $\vee$  SAY( $j, \wedge \forall v[\text{STUDENT}(v) \rightarrow \text{WIN}(v)]]$ )

<sup>16</sup> Except for the readings where quantifiers in conjuncts and disjuncts have wider scope than the conjunctions and disjunctions coordinating them. Those readings seem to be really impossible, cf. the discussion on the sentences (9) and (14) below.

‘While intuitions are far from clear’ (1983, p. 376), only (i) and (iii) are considered okay by Partee and Rooth, whereas in their fragment, (i) and (ii) are accounted for. A similar case is (149).

(149) Every student failed or got a D

This sentence does not seem to have the reading ‘every student failed or every student got a D’, a reading that seems to be predicted by all available theories of type ambiguity, including the present one: apply <sup>1</sup>VR and generalized disjunction to the intransitive verb phrases,  $\lambda T.[\forall T(\wedge \lambda x.FAIL(\forall x)) \vee \forall T(\wedge \lambda x.GET-D(\forall x))]$ , and combine this with the basic translation of *every student*. It is, however, obvious that the data are far from clear. Consider (150).

(150) Every player of our team is wearing a red shirt or a green shirt

That this sentence does have the reading ‘every player of our team is wearing a red shirt or every player of our team is wearing a green shirt’, can be supported by the possibility (if we add a ‘heavily loaded’ context: the speaker is colour-blind) of the continuation ‘... but I can’t tell you what the exact colour is’.

That the context must be heavily loaded is what we should expect. For, under normal circumstances the Gricean maxim ‘avoid ambiguity’ requires us not to utter (151), (152) or (153) to convey the information expressed by the ‘every ... or every ...’ readings; there are less ambiguous sentences that express them.<sup>17</sup>

(151) John said that every student lost or every student won

(152) Every student failed or every student got a D

(153) Every player of our team is wearing a red shirt or every player of our team is wearing a green shirt

Only a colour-blind person entangled in the cumbersomeness of his disjunctive world-view or a colour-blind person entangled in the cumbersomeness of her disjunctive world-view may be expected to let being short prevail over being unambiguous here. Besides, an explanation in terms of a pragmatic conflict between the Gricean maxims ‘be short’ and ‘avoid ambiguity’ can contribute to an explanation why reading (iii) for sentence

---

<sup>17</sup> The same pragmatic explanation applies to (135) (ii): there is an unambiguous sentence available for expressing this reading: *John caught a fish and ate a fish*. With respect to (148), *John said that every student lost or won*, Rooth and Partee (1982) note that in a parallel sentence like *John said that a student lost and won*, with *a* and *and* replacing *every* and *or*, the parallel wide scope-*and* reading is absolutely excluded: *John said that a student lost and won* cannot express ‘John said that a student lost and John said that a student won’. They attribute the difference to discourse properties of indefinite noun phrases. Something like this might very well be the case, for note that with a different noun phrase, the wide scope-*and* reading is not altogether impossible: *John said that exactly one student lost and won* does have the reading ‘John said that exactly one student lost and John said that exactly one student won’.

(148) = (151) is more plausible than reading (ii), whereas the former is semantically more ‘dislocated’ than the latter (i.e., it requires more type change) – a fact that otherwise would remain mysterious: the gain of time and saving of energy obtained by uttering *John said that every student lost or won* as ‘short for’ *John said that every student lost or John said that every student won* (reading (iii)), exceeds the profit made by using the former sentence instead of *John said that every student lost or every student won* (reading (ii)).<sup>18</sup>

In section 1.3.4 we noted that PTQ assigns the following translations to (9) and (14):

- (154) Mary walks and seeks every unicorn (= (9))  
 (i)  $WALK(m) \wedge SEEK(\lambda P. \forall v[UNICORN(v) \rightarrow \forall P(\lambda v)])(m)$   
 (ii)  $WALK(m) \wedge \forall v[UNICORN(v) \rightarrow SEEK(\lambda P. \forall P(\lambda v))(m)]$   
 (iii)  $\forall v[UNICORN(v) \rightarrow [WALK(m) \wedge SEEK(\lambda P. \forall P(\lambda v))(m)]]$   
 (155) John runs and no unicorn walks (= (14))  
 (i)  $RUN(j) \wedge \neg \exists v[UNICORN(v) \wedge WALK(v)]$   
 (ii)  $\neg RUN(j) \vee \neg \exists v[UNICORN(v) \wedge WALK(v)]$

The flexible grammar only generates the readings (154)(i), (ii), and (155)(i). Reading (154)(i) does not require any noteworthy type change. For (154)(ii) we change the type of *seek* via <sup>1</sup>AL and <sup>1</sup>AR, respectively:  $((s, e^2), ((s, e), t)) \Rightarrow^1AL ((s, e), ((s, e), t)) \Rightarrow^1AR ((s, e^2), ((s, e), t))$ , which leads to the derived translation

- (156)  $\lambda T \lambda y. \forall T(\lambda x. \wedge SEEK(\lambda P. \forall P(x))(\forall y))$ .

Application to the intension of the basic translation of *every unicorn* yields

- (157)  $\lambda y. \forall v[UNICORN(v) \rightarrow SEEK(\lambda P. \forall P(\lambda v))(\forall y)]$ ,

Generalized intransitive verb conjunction of (157) and  $\lambda y. WALK(\forall y)$  and application to the intension of *m* results in (ii). The generation of (154) (i) is straightforward. We shall not prove here that the non-readings (154) (iii) and (155) (ii) are not assigned within the flexible grammar (cf. the Appendix), but we shortly sketch why this is the case. We saw above that, anticipating the semantic structure in which this expression is embedded, a quantified noun phrase like *every unicorn* can be given wider scope than its original scope domain by applying value raising to the main functor of the scope domain (*seek* and *walk* in the cases under consideration), thus creating an ‘escape hatch’, and next letting it escape by applying argument raising to the relevant argument position. In the case of (9) ((14) is completely analogous) the semantic structure roughly looks like this: WALK + SEEK. We see that the left conjunct (WALK) is part of the semantic structure of

<sup>18</sup> This might also explain why the reading of the sentences in (28) with *or* having widest scope are more plausible than the readings where *or* has intermediate scope (that is, narrower than *claim that*, but wider than *seek*).



the sentence, but that the functor of the scope domain of *every unicorn*, SEEK, is not embedded in it. *Walk* ‘c-commands’ *seek*, but does not dominate this expression. This entails that there is no escape from *walk* – and reading (iii) cannot be constructed.<sup>19</sup>

### 4.3 Structural restrictions

The flexible grammar defined here is best seen as creating a ‘logical space’ of possibilities which is not necessarily exhausted by the facts of natural language. It should therefore meet the requirement that existing structural restrictions on possible readings can be implemented. In this section we shall show that this holds for scope islands and scope sieves.

It is well-known that some syntactic categories function as *scope islands*: quantified noun phrases or coordinations occurring inside expressions of island categories cannot have their scope outside of these environments. Relative clauses constitute a clear and relatively undisputed example of the phenomenon. Consider, e.g., (158) and (159).

(158) A girl who lives in every Italian city walks

(i)  $\exists u[\text{GIRL}(u) \wedge \forall v[[\text{ITALIAN-CITY}(v) \rightarrow \text{LIVE}(v)(u)] \wedge \text{WALK}(u)]]$

(ii)  $\forall v[\text{ITALIAN-CITY}(v) \rightarrow \exists u[[\text{GIRL}(u) \wedge \text{LIVE}(v)(u)] \wedge \text{WALK}(u)]]$

(159) Every girl who lives in Florence or Pisa walks

(i)  $\forall u[[\text{GIRL}(u) \wedge [\text{LIVE}(f)(u) \vee \text{LIVE}(p)(u)]] \rightarrow \text{WALK}(u)]$

(ii)  $\forall v[[\text{GIRL}(u) \wedge \text{LIVE}(f)(u)] \rightarrow \text{WALK}(u)] \vee \forall v[[\text{GIRL}(u) \wedge \text{LIVE}(p)(u)] \rightarrow \text{WALK}(u)]$

(158) can only have the (strange) interpretation (i); interpretation (ii) is impossible (though much more normal). Similarly, (159) can be interpreted as (i), but not as (ii).

Nonetheless, the sentences are each assigned two readings by the flexible grammar if we assume a (simplified) more or less standard treatment of relative clauses: (a) the relative pronoun *who* is assigned the basic translation  $\lambda P \lambda Q \lambda x. [\forall Q(x) \wedge \forall P(x)]$  of type  $((s, ((s, e), t)), ((s, ((s, e), t)), ((s, e), t)))$ ; (b) the basic type of relative clauses,  $t_B(\text{REL})$ , equals  $((s, ((s, e), t)), ((s, e), t))$ ; and (c) two more rules of application are added:

(xiii)  $\alpha_{\text{RELPRO}} + \beta_{\text{IV}} \rightarrow \alpha_0 = [\text{REL } \alpha_{\text{RELPRO}} \beta_{\text{IV}}]$   
 $\text{Tr}_B(\alpha_0) = \{(\text{FA}(\alpha', \beta') : \alpha' \in \text{Tr}(\alpha) \text{ and } \beta' \in \text{Tr}(\beta))\}$

<sup>19</sup> This is a consequence of our having separate rules of conjunction and disjunction. When *or* and *and* are treated as polymorphic functors (of syntactic category  $X \setminus (X/X)$  and semantic type  $(a, (a, a))$ , the usual analysis in flexible categorial grammars, cf. e.g., Moortgat (1988)), which combine with their arguments via functional application, the situation is different. Under that analysis, (154) (iii) and (155) (ii) are possible interpretations.

The analysis in terms of generalized conjunction and disjunction has the effect of turning coordinations into some kind of scope islands (cf. the treatment of relative clauses below), but, having scope themselves, they are ‘floating’ scope islands.

$$(xiv) \quad \beta_{CN+} \alpha_{REL} \rightarrow \alpha_0 = [CN \alpha_{CN} \beta_{REL}] \\ Tr_B(\alpha_0) = \{(FA(\alpha', \beta') : \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta))\}$$

We shall restrict ourselves to sentence (158). ((159) can be treated analogously.) Reading (i) is obtained by doing as little as possible: merely applying <sup>1</sup>AR to *live in*, and combining the result with the basic translation of *every Italian city* and the basic translation of *who*. This leads to the following translation of the relative clause *who lives in every Italian city*.

$$(160) \quad \lambda Q \lambda x. [\forall v [ITALIAN-CITY(v) \rightarrow LIVE(v)(\forall x)]]$$

In order to get reading (ii) of (158), we apply <sup>2</sup>VR and <sup>1</sup>AR (in that order) to  $\lambda x \lambda y. LIVE(\forall x)(\forall y)$ . The result is of type  $((s, e^2), e^3)$ :

$$(161) \quad \lambda T \lambda W. \forall T(\lambda x. \forall W(\lambda y. LIVE(\forall x)(\forall y)))$$

This is applied to  $\lambda P. \forall v [ITALIAN-CITY(v) \rightarrow \forall P(\forall v)]$ , and the result reduces to:

$$(162) \quad \lambda W. \forall v [ITALIAN-CITY(v) \rightarrow \forall W(\lambda y. LIVE(v)(\forall y))]$$

Now we let the basic translation of *who* undergo <sup>3</sup>AR and <sup>1</sup>AR, respectively, and apply the outcome,  $\lambda U \lambda Q \lambda T. \forall U(\lambda P. \forall T(\lambda x. [\forall Q(x) \wedge \forall P(x)]))$ , first to the intension of (162) and then to the basic translation of *girl*,  $\lambda x. [GIRL(\forall x) \wedge \exists u(x = \forall u)] = GIRL^*$ . Note that in this case the translation of the relative clause is not of the basic type  $t_B(REL) = ((s, ((s, e), t)), ((s, e), t))$ , but of the derived type  $((s, ((s, e), t)), ((s, e^2), t))$ . Thus, the compound common noun *girl who lives in every Italian city* is translated as:

$$(163) \quad [\lambda U \lambda Q \lambda T. \forall U(\lambda P. \forall T(\lambda x. [\forall Q(x) \wedge \forall P(x)]))] \\ (\lambda W. \forall v [ITALIAN-CITY(v) \rightarrow \forall W(\lambda y. LIVE(v)(\forall y))]) (\wedge GIRL^*) \Leftrightarrow \\ \lambda T. \forall v [ITALIAN-CITY(v) \rightarrow \forall T(\lambda x. [GIRL^*(x) \wedge LIVE(v)(\forall x)])]$$

Next, <sup>1</sup>AR is applied to the basic translation of the subject determiner *a*:  $\lambda B \lambda Q. \forall B(\lambda P. \exists x[\forall P(\forall x) \wedge \forall Q(\forall x)])$ ; this is combined with (163) and  $WALK^* = \lambda x. WALK(\forall x)$ .

$$(164) \quad [\lambda B \lambda Q. \forall B(\lambda P. \exists x[\forall P(x) \wedge \forall Q(x)])] \\ (\lambda T. \forall v [ITALIAN-CITY(v) \rightarrow \forall T(\lambda x. [GIRL^*(x) \wedge LIVE(v)(\forall x)]]) (\wedge WALK^*) \Leftrightarrow \\ \forall v [ITALIAN-CITY(v) \rightarrow \exists u [[GIRL(u) \wedge LIVE(v)(u)] \wedge WALK(u)]]$$

How is the fact that relative clauses constitute scope islands to be accounted for? The solution is very simple: we just add a condition which requires that all basic translations of relative clauses be of type  $t_B(REL) = ((s, ((s, e), t)), ((s, e), t))$ :

$$(165) \quad \text{For all } \alpha_{REL} \text{ and } \alpha' \in Tr_B(\alpha): TYPE(\alpha') = ((s, ((s, e), t)), ((s, e), t))$$

This is sufficient for turning relative clauses into barriers for quantification and coordination scope. Moreover, an interesting by-product of imposing (165) as a general restriction on the basic translations of relative clauses is that relative clauses are also predicted to be barriers for unbounded dependencies like topicalization and constituent question formation, witness (166) and (167).

- (166) \*Every Italian city a girl who loves walks  
 (167) \*Which Italian city a girl who loves walks?

In all semantically interpreted theories of unbounded dependencies, the semantic type assigned to gap-containing expressions of category  $\alpha$  differs (of course) from the type assigned to their gapless namesakes and, hence, from the basic type of  $\alpha$ ,  $t_B(\alpha)$ . In Gazdar, Klein, Pullum and Sag (1985, p. 230), for instance, an expression of category  $\alpha$  containing a gap of category  $\beta$  is assigned the type  $(TYPE(\beta), TYPE(a))$ . Therefore, condition (165) will rule out the existence of gap-containing relative clauses.<sup>21</sup>

Another kind of structural restriction on the availability of scope readings is the phenomenon of *scope sieves*: quantified noun phrases occurring inside expressions of sieve categories cannot have their scope at the level of the sieve. The existence of scope sieves has been pointed out by Asher and Bonevac (1985, 1987), Muskens (1989)<sup>21</sup> and Van der Does (1990). A sentence like (168), which contains a neutral perception verb (i.e., *see* plus a naked infinitival complement), cannot be interpreted as in (i). The embedded subject must have scope in the matrix sentence, cf. (ii).

- (168) John sees every man swim  
 (i)  $SEE(j, \wedge \forall v [MAN(v) \rightarrow SWIM(v)])$   
 (ii)  $\forall v [MAN(v) \rightarrow SEE(j, \wedge SWIM(v))]$

Note that we cannot characterize the correct readings by claiming that neutral perception verbs may not take scope over quantified noun phrases. E.g., in a perfectly acceptable *de dicto* reading of (169), (i), the verb *see* has scope over the determiner *a*. (169) also has a

<sup>20</sup> (Sequel to footnote 19.) We saw that an account using separate rules of generalized conjunction and disjunction has the effect of turning coordinations into ‘floating’ scope islands.

Given the observation about unbounded dependencies made here, we shall also predict the existence of ‘Across-the-Board’ phenomena (cf. Gazdar e.a. (1985), pp.176–178):

*The doctor who Kim worked for and Sandy relied on died*

\**The doctor who Kim worked for and Sandy relied on Lee died*

I.e., coordinations may contain gaps, but if they do, then both coordinated constituents must contain them. (The presence of only one gap would lead to different types of  $\alpha'$  and  $\beta'$ , and hence lead to inapplicability of generalized conjunction and disjunction.)

<sup>21</sup> In Muskens (1989), the bad readings are ruled out with the help of a semantic condition: the interpretation of the naked infinitival complement must be persistent. This works because syntactic variables are used in the representation of scope ambiguities, and these do not disturb the persistence of the interpretation of the complement.

correct *de re* reading, (ii). The intermediate reading (iii), however, in which *a pen* is interpreted *de dicto* with respect to *see*, but *de re* as regards *seek*, is out.

- (169) John sees Mary seek a pen  
 (i)  $SEE(j, \wedge SEEK(\wedge \lambda P. \exists u[PEN(u) \wedge \vee P(\wedge u)])(m))$   
 (ii)  $\exists u[PEN(u) \wedge SEE(j, \wedge SEEK(\wedge \lambda P. \vee P(\wedge u))(m))]$   
 (iii)  $SEE(j, \wedge \exists u[PEN(u) \wedge SEEK(\wedge \lambda P. \vee P(\wedge u))(m)])$

Within our flexible framework, the representation of this phenomenon is straightforward. The only assumption needed is that the naked infinitival forms of verbs which are translated as  $\alpha$  of type  $(\vec{a}, t)$ , are lexically translated as  $\lambda \vec{x}_a \lambda w_{(s, ((s, t), t))}. \vee w(\wedge [\alpha'(\vec{x})])$  of type  $(\vec{a}, t^2) = (\vec{a}, ((s, ((s, t), t)), t))$ . I.e., their basic translation looks like a VR term. (Maybe this is the right translation for verbal stems in general; the usual type  $(\vec{a}, t)$ , then, results from morphologically filling the last  $(s, ((s, t), t))$ -argument with tenses and aspects like *to*, *-ed* or *-ing*.)

For example, the naked infinitive *seek* is assigned the basic translation  $\lambda T \lambda y \lambda w. \vee w(\wedge SEEK(T)(\vee y))$ . If this is applied to the intension of  $\lambda P. \exists u[PEN(u) \wedge \vee P(\wedge u)]$  and  $m$ , respectively, and the result is combined with  $\lambda R \lambda x. \vee R(\wedge \lambda p. SEE(p)(\vee x))$  (the term obtained by raising the first argument of  $\lambda p \lambda x. SEE(p)(\vee x)$ , the basic translation of *see*) and the basic translation  $j$  of *John*, then reading (i) of sentence (169) is generated. If, in addition, the first argument of the naked infinitive translation is lowered and raised, we get (ii). It is impossible to obtain (iii). This approach also immediately accounts for principles like conjunction and disjunction distribution which have been claimed to hold for naked infinitives (as early as Barwise (1981)). In sentences like *John sees Mary walk and (or) Bill talk*, the ‘higher order’ translations of *Mary walk* and *Bill talk* will force the verb *sees* to distribute over the conjunction (disjunction), and hence *John sees Mary walk and (or) John sees Bill talk* will be entailed without further assumptions.

## 5. DISCUSSION: COMPOSITIONALITY AND CONTEXTUALITY

We have tried to show above that the adoption of flexible type assignment in Montague grammar leads to an empirically adequate semantic theory of quantification and coordination that can do without rules of quantification: the rules for flexible type assignment – value raising, argument raising and argument lowering – are empirically motivated. But one might also ask oneself whether there is a conceptual motivation for them: why these rules?

As for Montague’s rules of quantification, the situation is more or less clear. The principle of compositionality of meaning/reference entails that in all cases where a non-lexical ambiguity is found that cannot be reduced to different syntactic structures, a syntactic ambiguity needs to be ‘forced’. Montague implemented this as a derivational ambiguity: whenever an ambiguity cannot be accounted for in terms of the syntactic

structure (*de dicto/de re* and other scope ambiguities constitute clear examples), apparently different ways of constructing one and the same syntactic structure are at issue.

The grammar presented here is different in this respect. It is clear that non-lexical ambiguities do not have syntactic repercussions within this grammar: instead, one syntactic object is assigned a *set* of interpretations. However, in representing scope ambiguities no use is made of ‘artificial’ alternative ways of construction. We can restrict ourselves to the ‘intuitive’ syntactic structure, which is respected by all the syntactic operations of the fragment: constituent expressions are real *parts*, so the flexible grammar is even ‘more compositional’ than its predecessors (cf. Janssen (1983), pp. 65-66). Just as in PTQ, the meanings of a compound expression are determined by (i) the meanings of its constituent parts, and (ii) the semantic operation associated with the compounding syntactic operation, but with the difference that we can no longer speak about *the* meanings of the constituent parts. These expressions each have a *set* of interpretations, of which (and this is what matters) only the ‘mutually fitting’ are allowed to join (cf. the remark on partiality in section 3.2.2).

This is an important difference. We noted before that value raising can be used to anticipate, so to speak, the global semantic structure in which expressions are embedded. The same can be said with respect to argument raising and lowering, and the local semantic structure. These rules enable expressions to adapt themselves to the format of the arguments or functor they are combined with. In other words: besides the principle of compositionality, the principle of contextuality seems to play a role: ‘(...) nach der Bedeutung der Wörter muss in Satzzusammenhängen, nicht in ihrer Vereinzelnung gefragt werden (...)’ (Frege (1884), p. XXII).<sup>22</sup>

A strong interpretation of the principle of contextuality is involved here. Dummett discusses the status of the principle in Frege’s work, and arrives at what we might call a weak interpretation of the principle: ‘In a certain sense, sentences have a primacy within language over other linguistic expressions: a sentence is determined as true under certain conditions, which conditions are derivable from the way in which the sentence is constructed out of its constituent words; and the senses of the words relate solely to this determination of the truth conditions of the sentences in which the words may occur. Of course, looked at in one way, *the word has a sense independently of any particular sentence in which it occurs*: but its sense is something relating entirely to the occurrence of the word in a sentence (...).’ (Dummett (1976), pp.194-195, italics added). Under this weak interpretation (which, for that matter, seems to be the interpretation intended by Frege) the principle of contextuality is best considered a heuristic principle: in order to find out what *the* meaning of an expression *is* (‘independently of any particular sentence in which it occurs’), we must study the occurrence of that expression in the context of sentences. The fact that within the grammar of PTQ all expressions are assigned

---

<sup>22</sup> ‘The meaning of words can only be found in the context of sentences, not when they are studied in isolation.’

denotations of a relational type  $(\vec{a},t)$  illustrates the point: the denotation of expressions is identified with their contribution to the truth conditions of the sentences in which they occur.

The strong interpretation of the principle of contextuality explicitly denies the existence of the meaning of an expression, independent of the sentences in which that expression occurs. This is the interpretation which (in spite of himself) is expressed by Frege's statement 'Nur im Zusammenhange eines Satzes bedeuten die Wörter etwas.' (Frege (1884), p. 73).<sup>23</sup> In different sentences an expression can contribute in different ways to the truth conditions of the whole in which it occurs, and those contributions do not share a 'greatest common divisor'. In the flexible grammar presented above this is the case: take, for instance, the expression *finds*. In sentence (180), the contribution of this expression can be formalized as (181). In sentence (182), however, it contributes in two ways: (183) and (184).

(180) John finds Mary

(181)  $\lambda y \lambda x. \text{FIND}(\forall y)(\forall x)$

(182) Every man finds a woman

(183)  $\lambda T_1 \lambda T_2. \forall T_2(\wedge \lambda y. \forall T_1(\wedge \lambda x. \text{FIND}(\forall x)(\forall y)))$

(184)  $\lambda T_1 \lambda T_2. \forall T_1(\wedge \lambda x. \forall T_2(\wedge \lambda y. \text{FIND}(\forall x)(\forall y)))$

The fact that (181), the contribution to (180), is called the 'basic translation' in the grammar, is doubly misleading. First, this basic translation is not more essential or necessary than the derived translations in that it plays a role in all sentences in which the expression occurs: in (182), only derived translations are used. And second, in *all* cases where basic translations plays a role, this role could just as well be taken over by derived translations. For example, in the case of (180) we could proceed as in (185) or (186),

(185)  $\lambda T_1 \lambda T_2. \forall T_2(\wedge \lambda y. \forall T_1(\wedge \lambda x. \text{FIND}(\forall x)(\forall y))) (\wedge \lambda P. \forall P(\wedge m)) (\wedge \lambda P. \forall P(\wedge j))$

(186)  $\lambda T_1 \lambda T_2. \forall T_1(\wedge \lambda x. \forall T_2(\wedge \lambda y. \text{FIND}(\forall x)(\forall y))) (\wedge \lambda P. \forall P(\wedge m)) (\wedge \lambda P. \forall P(\wedge j)),$

but also in infinitely many different ways, which would without exception result in something equivalent to  $\text{FIND}(m)(j)$ .

---

<sup>23</sup> 'It is only in the context of a sentence that words have a meaning.'

## APPENDIX: SEMANTIC PROPERTIES OF THE FRAGMENT

*The fragment*

- (1)  $t_B(S): t$                        $t_B(TV): ((s,e),((s,e),t))$   
 $t_B(T): e$                                $t_B(PV): ((s,t),((s,e),t))$   
 $t_B(CN), t_B(IV): e^1$                $t_B(Det): ((s,e^1),e^2)$
- (2)  $T(C)$  is the smallest set such that  $t_B(C) \in T(C)$  and  $(\text{length}(\vec{a}) = i - 1)$ :  
 $[^iVR]$  if  $(\vec{a}, b) \in T(C)$ , then  $(\vec{a}, b^2) \in T(C)$ ;  
 $[^iAR]$  if  $(\vec{a}, ((s,b), (\vec{c}, t))) \in T(C)$ , then  $(\vec{a}, ((s,b^2), (\vec{c}, t))) \in T(C)$ ;  
 $[^iAL]$  if  $(\vec{a}, ((s,b^2), (\vec{c}, t))) \in T(C)$ , then  $(\vec{a}, ((s,b), (\vec{c}, t))) \in T(C)$ .
- (3)  $Tr(\alpha)$  is the smallest set such that  $Tr_B(\alpha) \subseteq Tr(\alpha)$  and  $(\text{length}(\vec{a}) = i - 1)$ :  
 $[^iVR]$  if  $\alpha' \in Tr(\alpha)$ , then  $\lambda \vec{x} \lambda w. \forall w (\wedge [\alpha'(\vec{x})]) \in Tr(\alpha)$ ;  
 $[^iAR]$  if  $\alpha' \in Tr(\alpha)$ , then  $\lambda \vec{x} \lambda w \lambda \vec{y}. \forall w (\wedge \lambda z. \alpha'(\vec{x})(z)(\vec{y})) \in Tr(\alpha)$ ;  
 $[^iAL]$  if  $\alpha' \in Tr(\alpha)$ , then  $\lambda \vec{x} \lambda w \lambda \vec{y}. \alpha'(\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y}) \in Tr(\alpha)$ .
- (4) For lexical expressions:  $Tr_B(\alpha) = \{tr_B(\alpha)_{\text{type}(tr_B(\alpha))}\}$ : cf. (84).
- (5)  $\gamma$  of type  $((s,a),b)$  and  $\delta$  of type  $a$ :  $FA(\gamma, \delta) = \gamma(\wedge \delta)$   
 $\gamma$  and  $\delta$  of type  $(\vec{a}, t)$ :  $GC(\gamma, \delta) = \lambda \vec{x}. [\gamma(\vec{x}) \wedge \delta(\vec{x})]$   
 $\gamma$  and  $\delta$  of type  $(\vec{a}, t)$ :  $GD(\gamma, \delta) = \lambda \vec{x}. [\gamma(\vec{x}) \vee \delta(\vec{x})]$
- (6)  $Tr_B(\alpha_0) = \{FA(\alpha', \beta'): \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$ :  
(i)  $\beta_T + \alpha_{IV} \rightarrow \alpha_0 = [s \beta_T \alpha_{IV}]$   
(ii)  $\alpha_{TV} + \beta_T \rightarrow \alpha_0 = [IV \alpha_{TV} \beta_T]$   
(iii)  $\alpha_{Det} + \beta_{CN} \rightarrow \alpha_0 = [T \alpha_{Det} \beta_{CN}]$   
(iv)  $\alpha_{PV} + \beta_S \rightarrow \alpha_0 = [IV \alpha_{PV} \beta_S]$   
 $Tr_B(\alpha_0) = \{GC(\alpha', \beta'): \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$ :  
(v)  $\alpha_S + \beta_S \rightarrow \alpha_0 = [s \alpha_S \text{ and } \beta_S]$   
(vi)  $\alpha_{IV} + \beta_{IV} \rightarrow \alpha_0 = [IV \alpha_{IV} \text{ and } \beta_{IV}]$   
(vii)  $\alpha_T + \beta_T \rightarrow \alpha_0 = [T \alpha_T \text{ and } \beta_T]$   
(viii)  $\alpha_{TV} + \beta_{TV} \rightarrow \alpha_0 = [TV \alpha_{TV} \text{ and } \beta_{TV}]$   
 $Tr_B(\alpha_0) = \{GD(\alpha', \beta'): \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$ :  
(ix)  $\alpha_S + \beta_S \rightarrow \alpha_0 = [s \alpha_S \text{ or } \beta_S]$   
(x)  $\alpha_{IV} + \beta_{IV} \rightarrow \alpha_0 = [IV \alpha_{IV} \text{ or } \beta_{IV}]$   
(xi)  $\alpha_T + \beta_T \rightarrow \alpha_0 = [T \alpha_T \text{ or } \beta_T]$   
(xii)  $\alpha_{TV} + \beta_{TV} \rightarrow \alpha_0 = [TV \alpha_{TV} \text{ or } \beta_{TV}]$

*Value raising  $\cap$  Argument raising*

VR and AR overlap. If  $(\vec{a}, b) \Rightarrow^{iVR} (\vec{a}, b^2)$  with  $b = ((s, c), t)$ , then the type transition can also be achieved using  $iAR$ :  $(\vec{a}, ((s, c), t)) \Rightarrow (\vec{a}, ((s, c^2), t)) = (\vec{a}, ((s, c), t)^2)$ . The translations assigned are equivalent:  $\lambda \vec{x} \lambda w. \forall w (\wedge [\alpha'(\vec{x})]) \Leftrightarrow \lambda \vec{x} \lambda w. \forall w (\wedge \lambda z. \alpha'(\vec{x})(z))$ . Therefore, we can at will assume that applications of VR are 'real VR', i.e.,  $b \neq ((s, c), t)$ .

*Argument lowering*

It can be shown that in the derivation of a translation all applications of AL can be ordered before AR and VR (a corollary of (I)); that AL can be assumed to have applied only to lexical translations (a corollary of the conjunction of (I) and (II)); and that the relative order of different AL-applications to a translation is semantically irrelevant (III).

- (I) (a)  $iAL(iVR(\alpha)) \Leftrightarrow \alpha$ ,  
 (b)  $iAL(iAR(\alpha)) \Leftrightarrow \alpha$ ,  
 (c) If  $i \neq j$ , then  $iAL(jVR(\alpha)) \Leftrightarrow jVR(iAL(\alpha))$ ,  
 (d) If  $i \neq j$ , then  $iAL(jAR(\alpha)) \Leftrightarrow jAR(iAL(\alpha))$ .

(a)  $iAL$  is applied immediately after  $iVR$ : then  $\text{type}(iVR(\alpha)) = (\vec{a}, ((s, c^2), t))$ , with  $\text{length}(\vec{a}) = i-1$ . So  $iVR$  is not an instance of real VR, and the case reduces to case (b):

(b)  $iAL$  is applied immediately after  $iAR$ :  $\text{type}(iAR(\alpha)) = (\vec{a}, ((s, b^2), (\vec{c}, t)))$ , and  $\text{type}(\alpha) = \text{type}(iAL(iAR(\alpha))) = (\vec{a}, ((s, b), (\vec{c}, t)))$ , with  $\text{length}(\vec{a}) = i-1$ . Prove that  $(iAL(iAR(\alpha))) \Leftrightarrow \alpha$  by  $\lambda$ -conversion and  $\forall$ -elimination:

$$iAL(iAR(\alpha)) = \lambda \vec{x} \lambda w \lambda \vec{y}. [\lambda \vec{x}' \lambda w' \lambda \vec{y}'. \forall w' (\wedge \lambda z'. \alpha'(\vec{x}')(z')(\vec{y}'))](\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y}) \Leftrightarrow \alpha.$$

(c)  $iAL$  is applied immediately after  $jVR$ ,  $i \neq j$ :  $\text{type}(jVR(\alpha)) = (\vec{x}, d^2)$ ,  $\text{length}(\vec{x}) = j-1$ . Note that  $i < j$ , for the type  $(\vec{x}, d^2)$  has only  $\text{length}(\vec{x})+1 = j$  arguments. Hence  $(\vec{x}, d^2) = (\vec{a}, ((s, b^2), (\vec{c}, d^2)))$  and  $\text{type}(\alpha) = \text{type}(iAL(jVR(\alpha))) = (\vec{a}, ((s, b), (\vec{c}, d^2)))$ .

Prove that  $iAL(jVR(\alpha)) \Leftrightarrow jVR(iAL(\alpha))$  by  $\lambda$ -conversion and  $\forall$ -elimination.

(d)  $iAL$  is applied immediately after  $jAR$ ,  $i \neq j$ . Distinguish the cases  $i > j$  and  $i < j$ . Prove that  $iAL(jAR(\alpha)) \Leftrightarrow jAR(iAL(\alpha))$  by  $\lambda$ -conversion and  $\forall$ -elimination.

- (II) (a)  $iAL(FA(\alpha, \beta)) = FA(i^{i+1}AL(\alpha), \beta)$   
 (b)  $iAL(GC(\alpha, \beta)) = GC(iAL(\alpha), iAL(\beta))$   
 (c)  $iAL(GD(\alpha, \beta)) = GD(iAL(\alpha), iAL(\beta))$

Proofs by  $\lambda$ -conversion and  $\forall$ -elimination. E.g. (b):  $iAL(GC(\alpha, \beta)) = \lambda \vec{x} \lambda w \lambda \vec{y}. [\lambda \vec{x}' \lambda w' \lambda \vec{y}'. [\alpha(\vec{x}')(w')(\vec{y}') \wedge \beta(\vec{x}')(w')(\vec{y}')]](\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y}) \Leftrightarrow \lambda \vec{x} \lambda w \lambda \vec{y}. [\alpha(\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y}) \wedge \beta(\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y})]$ , and  $GC(iAL(\alpha), iAL(\beta)) = \lambda \vec{x} \lambda w \lambda \vec{y}. [[\lambda \vec{x}' \lambda w' \lambda \vec{y}'. \alpha(\vec{x}')(w')(\vec{y}')](\vec{x})(w)(\vec{y}) \wedge [\lambda \vec{x}' \lambda w' \lambda \vec{y}'. \beta(\vec{x}')(w')(\vec{y}')]](\vec{x})(w)(\vec{y})] \Leftrightarrow \lambda \vec{x} \lambda w \lambda \vec{y}. [\alpha(\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y}) \wedge \beta(\vec{x})(\wedge \lambda z. \forall z(w))(\vec{y})]$

- (III)  $iAL(jAL(\alpha)) \Leftrightarrow jAL(iAL(\alpha))$ .

Prove that  $iAL(jAL(\alpha)) \Leftrightarrow jAL(iAL(\alpha))$  by  $\lambda$ -conversion and  $\forall$ -elimination.



*Combinability of types*

(IV) If types  $a'$ ,  $f' = ((s,a'),b)$  have been derived from types  $a$ ,  $f$ , respectively, by means of VR and AR, then (i) the derivation  $a \Rightarrow a'$  does not contain any real  ${}^1\text{VR}$ , or (ii) the derivation  $f \Rightarrow f'$  does not contain any real  ${}^1\text{VR}$ .

Proof: Suppose both derivations contain real  ${}^1\text{VR}$ . Consider the *last* real  ${}^1\text{VR}$  applied in the derivation  $a \Rightarrow a'$ :  $a_j \Rightarrow {}^1\text{VR } a_{j+1} = a_j^2$ . Since  ${}^1\text{VR}$  is real,  $a_j$  is *not* of the form  $((s,x),t)$ . After that, only  ${}^i\text{VR}$ ,  ${}^i\text{AR}$  ( $i > 1$ ) and  ${}^1\text{AR}$  ( $p$  times,  $p \in \mathbb{N}$ ) have applied, so  $a' = ((s,a_j^{2p+1}),w)$ . Consider the *last* real  ${}^1\text{VR}$  applied in  $f \Rightarrow f'$ :  $f_k \Rightarrow {}^1\text{VR } f_{k+1} = f_k^2$ . Since  ${}^1\text{VR}$  is real,  $f_k$  is *not* of the form  $((s,x),t)$ . After that, only  ${}^i\text{VR}$ ,  ${}^i\text{AR}$  ( $i > 1$ ) and  ${}^1\text{AR}$  ( $q$  times) have applied, so  $f' = ((s,f_k^{2q+1}),y)$ . Since  $f_k^{2q+1} = a_n = ((s,a_j^{2p+1}),w)$ , we know that  $w = t$ , and  $f_k^{2q} = a_j^{2p+1}$ . But then it must hold that either  $f_k$  (if  $q \leq p$ ) or  $a_j$  (if  $q > p$ ) is of the form  $((s,x),t)$ . Contradiction.

(V) If types  $a'$ ,  $f' = ((s,a'),c)$  have been derived from types  $x$ ,  $((s,x),b)$  [call such types 'fitting'], respectively, by VR and AR, then  $f \Rightarrow f'$  does not contain real  ${}^1\text{VR}$ .

Proof: If  $x \Rightarrow a'$  contains real  ${}^1\text{VR}$ , then  $f \Rightarrow f'$  does not contain real  ${}^1\text{VR}$  (use (IV)). So, assume that  $x \Rightarrow a'$  does not contain real  ${}^1\text{VR}$ , but that  $f \Rightarrow f'$  does. Note that the latter entails that  $a' = ((s,z),t)$ . Now, if  $x \Rightarrow a'$  contains  ${}^i\text{VR}$ ,  ${}^i\text{AR}$  (for  $i > 1$ ) and no  ${}^1\text{VR}$ , we have that  $a' \neq ((s,z),t)$ . Hence  $x \Rightarrow a'$  does not contain real  ${}^1\text{VR}$  and  ${}^i\text{VR}$ ,  ${}^i\text{AR}$  for  $i > 1$ , but only  ${}^1\text{AR}$  ( $r$  times), leading to a type  $((s,z),t)$ , so  $a' = ((s,x^{2r}),t)$ . On the other hand, the first real  ${}^1\text{VR}$  (after  $q$  times  ${}^1\text{AR}$ ) in  $f = ((s,x),b) \Rightarrow f' = ((s,a'),c)$  entails that  $a'$  has  $((s,((s,x^{2q}),d)),t)$  (where  $d \neq t$ ) as a subtype. Contradiction.

(VI) If expression  $\alpha$  of category  $C$  translates as  $\alpha'$  of type  $a'$  [ $\alpha'_a$ ], then  $t_B(C) \Rightarrow a'$ .

Proof by induction on the complexity of  $\alpha$ : (i) the claim is trivial for lexical expressions. (ii) GC/GD:  $\alpha_C = [C\beta_C \text{ and } \gamma_C]$  or  $\alpha_C = [C\beta_C \text{ or } \gamma_C]$  and  $\alpha'_a$  is derived from GC/GD( $\beta', \gamma'$ ) $_a$ :  $a \Rightarrow a'$ . By induction hypothesis (IH):  $t_B(C) \Rightarrow \text{type}(\beta')$  and  $t_B(C) \Rightarrow \text{type}(\gamma')$ . Note that  $\text{type}(\beta') = \text{type}(\gamma') = a$ . So  $t_B(C) \Rightarrow a'$ . (iii) FA: distinguish four cases. (a)  $\alpha_S = [s\gamma_T \beta_{IV}]$  and  $\alpha'_a$  is derived from FA( $\beta', \gamma'$ ) $_a$ :  $a \Rightarrow a'$ . IH:  $t_B(IV) = ((s,e),t) \Rightarrow \text{type}(\beta') = ((s,\text{type}(\gamma'),a)$  and  $t_B(T) = e \Rightarrow \text{type}(\gamma')$ ; (b)  $\alpha_{IV} = [IV\beta_{TV} \gamma_T]$  and  $\alpha'_a$  is derived from FA( $\beta', \gamma'$ ) $_a$ :  $a \Rightarrow a'$ . IH:  $t_B(TV) = ((s,e),((s,e),t)) \Rightarrow \text{type}(\beta') = ((s,\text{type}(\gamma'),a)$  and  $t_B(T) = e \Rightarrow \text{type}(\gamma')$ ; (c)  $\alpha_T = [T\beta_{Det} \gamma_{CN}]$  and  $\alpha'_a$  is derived from FA( $\beta', \gamma'$ ) $_a$ :  $a \Rightarrow a'$ . IH:  $t_B(Det) = ((s,((s,e),t)),((s,((s,e),t)),t)) \Rightarrow \text{type}(\beta') = ((s,\text{type}(\gamma'),a)$  and  $t_B(CN) = e^1 \Rightarrow \text{type}(\gamma')$ ; (d)  $\alpha_{IV} = [IV\beta_{PV} \gamma_S]$  and  $\alpha'_a$  is derived from FA( $\beta', \gamma'$ ) $_a$ :  $a \Rightarrow a'$ . IH:  $t_B(PV) = ((s,t),((s,e),t)) \Rightarrow \text{type}(\beta') = ((s,\text{type}(\gamma'),a)$  and  $t_B(S) = t \Rightarrow \text{type}(\gamma')$ . Note that in (a) through (d) (V) can be applied: the types  $t_B(C_\beta)$  and  $t_B(C_\gamma)$  are fitting [and not subject to  ${}^1\text{AL}$ ]:  $t_B(C_\beta) = ((s,t_B(C_\gamma)),d)$ , so  $t_B(C_\beta) \Rightarrow \text{type}(\beta') = ((s,\text{type}(\gamma'),a)$  contains no  ${}^1\text{VR}$ . But this entails that  $d \Rightarrow a$  [since the application of  ${}^1\text{AR}$  to a type  $((s,x),y)$  does not affect  $y$ ], and hence  $d \Rightarrow a'$ . Thus: (a)  $d = t = t_B(S) \Rightarrow a'$ ; (b)  $d = ((s,e),t) = t_B(IV) \Rightarrow a'$ ; (c)  $e^2 = ((s,((s,e),t)),t) = d \Rightarrow a'$  [and  $e = t_B(T) \Rightarrow {}^1\text{VR } e^2$ ]; and (d)  $d = ((s,e),t) = t_B(IV) \Rightarrow a'$ .

Corollary: apart from AL, the fact that the fragment is fitting entails that the argument categories of the fragment [S,T,CN] only undergo  ${}^1\text{VR}$ ; that unary non-main functor categories [IV,Det] only undergo  ${}^1\text{AR}$  and  ${}^2\text{VR}$ ; and that binary non-main functor categories [TV, PV] only undergo  ${}^1\text{AR}$ ,  ${}^2\text{AR}$ , and  ${}^3\text{VR}$ . Moreover, observe that  $c \Rightarrow t$  iff  $c = t$  (all type change leads to compound types), so in sentence readings [interpretations of type t, cf. (106) above] unary main functor categories [IV,Det] only undergo  ${}^1\text{AR}$ , and binary main functor categories [TV,PV] only undergo  ${}^1\text{AR}$  and  ${}^2\text{AR}$ .

### *Value raising and argument raising*

Applications of AR and VR within functional application structures can be shifted back to the functor. It follows that in pure application structures (structures which contain no generalized conjunction and disjunction) only lexical type change is needed.

$$\begin{aligned} \text{(VII)} \quad & \text{(a) } {}^i\text{AR}(\text{FA}(\alpha, \beta)) \Leftrightarrow \text{FA}({}^{i+1}\text{AR}(\alpha), \beta) \\ & \text{(b) } {}^i\text{VR}(\text{FA}(\alpha, \beta)) \Leftrightarrow \text{FA}({}^{i+1}\text{VR}(\alpha), \beta) \end{aligned}$$

Proof: (a)  ${}^i\text{AR}(\text{FA}(\alpha, \beta)) = \lambda \vec{x} \lambda w \lambda \vec{y}. \forall w (\wedge \lambda z. \alpha(\wedge \beta)(\vec{x})(z)(\vec{y}))$ . Note that since  $\alpha \Leftrightarrow \lambda v \lambda \vec{x} \lambda w \lambda \vec{y}. \alpha(v)(\vec{x})(w)(\vec{y})$ , we have  ${}^{i+1}\text{AR}(\alpha) \Leftrightarrow \lambda v \lambda \vec{x} \lambda w \lambda \vec{y}. \forall w (\wedge \lambda z. \alpha(v)(\vec{x})(z)(\vec{y}))$ , and  $\text{FA}({}^{i+1}\text{AR}(\alpha), \beta) = {}^{i+1}\text{AR}(\alpha)(\wedge \beta) \Leftrightarrow [\lambda v \lambda \vec{x} \lambda w \lambda \vec{y}. \forall w (\wedge \lambda z. \alpha(v)(\vec{x})(z)(\vec{y}))](\wedge \beta) \Leftrightarrow \lambda \vec{x} \lambda w \lambda \vec{y}. \forall w (\wedge \lambda z. \alpha(\wedge \beta)(\vec{x})(z)(\vec{y}))$ . (b) is analogous to (a).

When translations  $\varphi(\vec{a}, ((s, b^2), d))$  and  $\alpha(\vec{c}, b^2)$  of a functor and (the  $j-1$ -ary main functor) of its  $i$ -th argument are such that a *final*  $j\text{VR}$ -application to (the main functor of) that argument ‘answers’ an  ${}^i\text{AR}$ -application to the functor, both applications can be omitted.

(VIII) If  $\varphi = X_1(\dots(X_n({}^i\text{AR}(\varphi_0)))\dots)$  [ $X_j = p\text{VR}$ ,  $p > i$  or  $q\text{AR}$ ,  $q \neq i$ ] and  $\alpha = {}^i\text{VR}(\alpha_0)$ , then for  $\varphi' = X_1(\dots(X_n(\varphi_0))\dots)$ :  $\varphi(\vec{\psi})(\wedge \alpha(\vec{\chi})) \Leftrightarrow \varphi'(\vec{\psi})(\wedge \alpha_0(\vec{\chi}))$  [length( $\vec{\psi}$ )= $i-1$ , length( $\vec{\chi}$ )= $j-1$ ]. ([...] will be used for gaps in contexts,  $\langle \dots \rangle$  for substitutions.)

Proof: Note that  $\alpha \Leftrightarrow \lambda \vec{z} \lambda v. \forall v (\wedge \alpha_0(\vec{z}))$ ,  $\varphi \Leftrightarrow \lambda \vec{x} \lambda w \lambda \vec{y}. \Phi[\forall w (\wedge \lambda w'. \varphi_0)]$ , and  $\varphi' \Leftrightarrow \lambda \vec{x} \lambda w' \lambda \vec{y}. \Phi[\varphi_0]$ , where  $\vec{z}$  does not occur in  $\alpha_0$  and  $w'$  does not occur in  $\Phi$ .

$$\begin{aligned} \varphi(\vec{\psi})(\wedge \alpha(\vec{\chi})) & \Leftrightarrow \lambda \vec{x} \lambda w \lambda \vec{y}. \Phi[\forall w (\wedge \lambda w'. \varphi_0)](\vec{\psi})(\wedge \lambda \vec{z} \lambda v. \forall v (\wedge \alpha_0(\vec{z}))(\vec{\chi})) \Leftrightarrow \\ & \lambda w \lambda \vec{y}. \Phi[\forall w (\wedge \lambda w'. \varphi_0)] \langle \vec{x} := \vec{\psi} \rangle (\wedge \lambda v. \forall v (\wedge \alpha_0(\vec{\chi}))) \Leftrightarrow \\ & \lambda \vec{y}. \Phi[\lambda v. \forall v (\wedge \alpha_0(\vec{\chi})) (\wedge \lambda w'. \varphi_0)] \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow \lambda \vec{y}. \Phi[\lambda w'. \varphi_0 (\wedge \alpha_0(\vec{\chi}))] \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow \\ & \lambda \vec{y}. \Phi[\varphi_0 \langle w' := \wedge \alpha_0(\vec{\chi}) \rangle] \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow \lambda \vec{y}. \Phi[\varphi_0] \langle w' := \wedge \alpha_0(\vec{\chi}) \rangle \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow \\ & \lambda \vec{x} \lambda w' \lambda \vec{y}. \Phi[\varphi_0](\vec{\psi})(\wedge \alpha_0(\vec{\chi})) \Leftrightarrow \varphi'(\vec{\psi})(\wedge \alpha_0(\vec{\chi})). \end{aligned}$$

When translations  $\varphi(\vec{a}, ((s, b^4), d))$  and  $\alpha(\vec{c}, b^4)$  of a functor and (the  $j-1$ -ary main functor of) its  $i$ -th argument are such that two *consecutive* applications of  $j\text{VR}$  to (the main functor of) that argument ‘answer’ two – possibly non-consecutive – applications or  ${}^i\text{AR}$  to the functor, we can omit one  $j\text{VR}$  and the ‘innermost’  ${}^i\text{AR}$ . [Similarly in the case where translations  $\varphi(\vec{a}, ((s, b^4), d))$  and  $\alpha(\vec{c}, b^4)$  of a functor and (the  $j-1$ -ary main functor of) its  $i$ -th argument contain two consecutive applications of  ${}^i\text{AR}$  to the functor which ‘answer’ two – possibly non-consecutive – applications or  $j\text{VR}$  to (the main functor of) that argument. There we can skip one  ${}^i\text{AR}$  and the ‘outermost’  $j\text{VR}$ . Proof omitted.]

(IX) If  $\varphi = X_1(\dots(X_k({}^i\text{AR}(X_{k+1}(\dots(X_n({}^i\text{AR}(\varphi_0))))))\dots)) [X_1\dots X_n = \text{AR or PVR } (p > i)]$  and  $\alpha = Y_1(\dots(Y_m({}^j\text{VR}({}^j\text{VR}(\alpha_0))))\dots) [Y_1\dots Y_m = \text{AR or } {}^q\text{VR } (q \geq j)]$ , where  $\#({}^i\text{AR in } X_{k+1}\dots X_n) = 0$  and  $\#({}^i\text{AR in } X_1\dots X_k) = \#({}^j\text{VR}+{}^j\text{AR in } Y_1\dots Y_m) = r$ , then for  $\varphi' = X_1(\dots(X_i({}^i\text{AR}(X_{i+1}(\dots(X_n(\varphi_0))))))\dots)$  and  $\alpha' = Y_1(\dots(Y_m({}^j\text{VR}(\alpha_0))))\dots$ :  $\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow \varphi'(\vec{\psi})(\wedge\alpha'(\vec{\chi}))$  [length( $\vec{\psi}$ ) =  $i-1$ , length( $\vec{\chi}$ ) =  $j-1$ ].

Proof by induction on  $r$ .

$r=0$ : Note that  $\varphi \Leftrightarrow \lambda\vec{x}\lambda v\lambda\vec{y}.\Phi_1[\vee v(\wedge\lambda v'.\Phi_2[\vee v'(\wedge\lambda v''.\varphi_0)])]$ , and that  $\varphi' \Leftrightarrow \lambda\vec{x}\lambda v'\lambda\vec{y}.\Phi_1[\vee v'(\wedge\lambda v''.\Phi_2[\varphi_0])]$ , where  $v''$  does not occur in  $\Phi_2$ . Moreover,  $\alpha \Leftrightarrow \lambda\vec{z}\lambda f.A[\vee f(\wedge\lambda f'.\vee f'(\wedge\lambda f''.\alpha_0))]$  and  $\alpha' \Leftrightarrow \lambda\vec{z}\lambda f'.A[\vee f'(\wedge\lambda f''.\alpha_0)]$ .

$\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow$   
 $\lambda v\lambda\vec{y}.\Phi_1[\vee v(\wedge\lambda v'.\Phi_2[\vee v'(\wedge\lambda v''.\varphi_0)])] \langle \vec{x} := \vec{\psi} \rangle (\wedge\lambda f.A[\vee f(\wedge\lambda f'.\vee f'(\wedge\lambda f''.\alpha_0))] \langle \vec{z} := \vec{\chi} \rangle)$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[\lambda f.A[\vee f(\wedge\lambda f'.\vee f'(\wedge\lambda f''.\alpha_0))] \langle \vec{z} := \vec{\chi} \rangle (\wedge\lambda v'.\Phi_2[\vee v'(\wedge\lambda v''.\varphi_0)])] \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[A[\lambda v'.\Phi_2[\vee v'(\wedge\lambda v''.\varphi_0)](\wedge\lambda f'.\vee f'(\wedge\lambda f''.\alpha_0))] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[A[\Phi_2[\lambda f'.\vee f'(\wedge\lambda f''.\alpha_0)](\wedge\lambda v''.\varphi_0)] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[A[\Phi_2[\varphi_0 \langle v'' := \wedge\lambda f''.\alpha_0 \rangle]] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[A[\Phi_2[\varphi_0] \langle v'' := \wedge\lambda f''.\alpha_0 \rangle] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[A[\lambda v''.\Phi_2[\varphi_0](\wedge\lambda f''.\alpha_0)] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi_1[\lambda f'.A[\vee f'(\wedge\lambda f''.\alpha_0)] \langle \vec{z} := \vec{\chi} \rangle (\wedge\lambda v''.\Phi_2[\varphi_0])] \langle \vec{x} := \vec{\psi} \rangle$   
 $\Leftrightarrow \lambda v'\lambda\vec{y}.\Phi_1[\vee v'(\wedge\lambda v''.\Phi_2[\varphi_0])] \langle \vec{x} := \vec{\psi} \rangle (\wedge\lambda f'.A[\vee f'(\wedge\lambda f''.\alpha_0)] \langle \vec{z} := \vec{\chi} \rangle)$   
 $\Leftrightarrow \varphi'(\vec{\psi})(\wedge\alpha'(\vec{\chi})).$

$r>0$ : Since  $r > 0$ , we can assume that  $\varphi \Leftrightarrow \lambda\vec{x}\lambda v\lambda\vec{y}.\Phi[\vee v(\wedge\lambda v'.\varphi_0(\vec{x}')(v')(\vec{y}'))]$  and that  $\alpha \Leftrightarrow \lambda\vec{z}\lambda w.A[\vee w(\wedge\alpha_0(\vec{z}'))]$  [length( $\vec{x}'$ ) =  $i-1$ , length( $\vec{z}'$ ) =  $j-1$ ]. Note that  $\varphi_0$  and  $\alpha_0$  are derived translations that have undergone  $r$  times  ${}^i\text{AR}$  and  ${}^j\text{VR}$ , respectively. Hence the induction hypothesis is applicable to  $\varphi_0$  and  $\alpha_0$ : there are simpler terms  $\varphi_0'$  and  $\alpha_0'$  (with  $r-1$  times  ${}^i\text{AR}$  and  ${}^j\text{VR}$ , respectively) such that:  $\varphi_0(\vec{x}')(\wedge\alpha_0(\vec{z}')) \Leftrightarrow \varphi_0'(\vec{x}')(\wedge\alpha_0'(\vec{z}'))$ . But then for  $\varphi'' = \lambda\vec{x}\lambda v\lambda\vec{y}.\Phi[\vee v(\wedge\lambda v'.\varphi_0'(\vec{x}')(v')(\vec{y}'))]$ ,  $\alpha'' = \lambda\vec{z}\lambda w.A[\vee w(\wedge\alpha_0'(\vec{z}'))]$  it will hold that  $\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow \varphi''(\vec{\psi})(\wedge\alpha''(\vec{\chi}))$ :

$\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow$   
 $\lambda v\lambda\vec{y}.\Phi[\vee v(\wedge\lambda v'.\varphi_0(\vec{x}')(v')(\vec{y}'))] \langle \vec{x} := \vec{\psi} \rangle (\wedge\lambda w.A[\vee w(\wedge\alpha_0(\vec{z}'))] \langle \vec{z} := \vec{\chi} \rangle) \Leftrightarrow$   
 $\lambda\vec{y}.\Phi[\lambda w.A[\vee w(\wedge\alpha_0(\vec{z}'))] \langle \vec{z} := \vec{\chi} \rangle (\wedge\lambda v'.\varphi_0(\vec{x}')(v')(\vec{y}'))] \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow$   
 $\lambda\vec{y}.\Phi[A[\lambda v'.\varphi_0(\vec{x}')(v')(\vec{y}')](\wedge\alpha_0(\vec{z}'))] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow$   
 $\lambda\vec{y}.\Phi[A[\varphi_0(\vec{x}')(\wedge\alpha_0(\vec{z}'))(\vec{y}')] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow \text{(IH)}$   
 $\lambda\vec{y}.\Phi[A[\varphi_0'(\vec{x}')(\wedge\alpha_0'(\vec{z}'))(\vec{y}')] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow$   
 $\lambda\vec{y}.\Phi[A[\lambda v'.\varphi_0'(\vec{x}')(v')(\vec{y}')](\wedge\alpha_0'(\vec{z}'))] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow$   
 $\lambda\vec{y}.\Phi[\lambda w.A[\vee w(\wedge\alpha_0'(\vec{z}'))] \langle \vec{z} := \vec{\chi} \rangle (\wedge\lambda v'.\varphi_0'(\vec{x}')(v')(\vec{y}'))] \langle \vec{x} := \vec{\psi} \rangle \Leftrightarrow$   
 $\lambda v\lambda\vec{y}.\Phi[\vee v(\wedge\lambda v'.\varphi_0'(\vec{x}')(v')(\vec{y}'))] \langle \vec{x} := \vec{\psi} \rangle (\wedge\lambda w.A[\vee w(\wedge\alpha_0'(\vec{z}'))] \langle \vec{z} := \vec{\chi} \rangle) \Leftrightarrow$   
 $\varphi''(\vec{\psi})(\wedge\alpha''(\vec{\chi})).$

*Generalized conjunction and disjunction*

(VIII) generalizes to cases where the translation  $\varphi$  of the functor consists of (possibly iterated) *coordinations* (conjunctions and/or disjunctions) of translations  $\varphi_1, \dots, \varphi_s$ : when translations  $\varphi(\vec{a}, ((s, b^2), d))$  and  $\alpha(\vec{c}, b^2)$  of a functor with coordinates  $\{\varphi_1, \dots, \varphi_s\}$  and (the  $j-1$ -ary main functor) of its  $i$ -th argument are such that a *final*  $j$ VR-application to (the main functor of) that argument ‘answers’  $i$ AR-applications to all coordinates of the functor, these applications can be omitted.

(X) If  $\varphi = X_1(\dots(X_n(iAR(\{\varphi_1, \dots, \varphi_s\})))\dots)$  [ $X_j = GD, GC, PVR, p > i$  or  $qAR, q \neq i$ ] and  $\alpha = iVR(\alpha_0)$ , then for  $\varphi' = X_1(\dots(X_n(\{\varphi_1, \dots, \varphi_s\})))\dots$ :  $\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow \varphi'(\vec{\psi})(\wedge\alpha_0(\vec{\chi}))$  [ $\text{length}(\vec{\psi})=i-1, \text{length}(\vec{\chi}) = j-1$ ].

Proof:  $\varphi \Leftrightarrow \lambda\vec{x}\lambda w\lambda\vec{y}.\Phi[\vee w(\wedge\lambda w'.\varphi_1), \dots, \vee w(\wedge\lambda w'.\varphi_s)], \alpha \Leftrightarrow \lambda\vec{z}\lambda v.\vee v(\wedge\alpha_0(\vec{z}))$ , and  $\varphi' \Leftrightarrow \lambda\vec{x}\lambda w\lambda\vec{y}.\Phi[\varphi_1, \dots, \varphi_s]$ , where  $\vec{z}$  does not occur in  $\alpha_0$  and  $w'$  does not occur in  $\Phi$ .  
 $\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow \lambda\vec{x}\lambda w\lambda\vec{y}.\Phi[\vee w(\wedge\lambda w'.\varphi_1), \dots, \vee w(\wedge\lambda w'.\varphi_s)](\vec{\psi})(\wedge\lambda\vec{z}\lambda v.\vee v(\wedge\alpha_0(\vec{z}))(\vec{\chi}))$   
 $\Leftrightarrow \lambda w\lambda\vec{y}.\Phi[\vee w(\wedge\lambda w'.\varphi_1), \dots, \vee w(\wedge\lambda w'.\varphi_s)]\langle\vec{x}:=\vec{\psi}\rangle(\wedge\lambda v.\vee v(\wedge\alpha_0(\vec{\chi})))$   
 $\Leftrightarrow \lambda\vec{y}.\Phi[\lambda v.\vee v(\wedge\alpha_0(\vec{\chi}))(\wedge\lambda w'.\varphi_1), \dots, \lambda v.\vee v(\wedge\alpha_0(\vec{\chi}))(\wedge\lambda w'.\varphi_s)]\langle\vec{x}:=\vec{\psi}\rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi[\lambda w'.\varphi_1(\wedge\alpha_0(\vec{\chi})), \dots, \lambda w'.\varphi_s(\wedge\alpha_0(\vec{\chi}))]\langle\vec{x}:=\vec{\psi}\rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi[\varphi_1\langle w':=\wedge\alpha_0(\vec{\chi})\rangle, \dots, \varphi_s\langle w':=\wedge\alpha_0(\vec{\chi})\rangle]\langle\vec{x}:=\vec{\psi}\rangle$   
 $\Leftrightarrow \lambda\vec{y}.\Phi[\varphi_1, \dots, \varphi_s]\langle w':=\wedge\alpha_0(\vec{\chi})\rangle\langle\vec{x}:=\vec{\psi}\rangle$   
 $\Leftrightarrow \lambda\vec{x}\lambda w\lambda\vec{y}.\Phi[\varphi_1, \dots, \varphi_s](\vec{\psi})(\wedge\alpha_0(\vec{\chi})) \Leftrightarrow \varphi'(\vec{\psi})(\wedge\alpha_0(\vec{\chi}))$ .

(IX) generalizes to cases where both translations  $\varphi$  and  $\alpha$  of a functor and (the  $j-1$ -ary main functor of) its  $i$ -th argument are (possibly iterated) *coordinations* of translations  $\varphi_1, \dots, \varphi_s$  and  $\alpha_1, \dots, \alpha_t$ : when translations  $\varphi(\vec{a}, ((s, b^4), d))$  and  $\alpha(\vec{c}, b^4)$  with coordinates  $\{\varphi_1, \dots, \varphi_s\}$  and  $\{\alpha_1, \dots, \alpha_t\}$ , respectively, of a functor and (the  $j-1$ -ary main functor of) its  $i$ -th argument are such that two *consecutive* applications of  $j$ VR to all coordinates of (the main functor of) that argument ‘answer’ two – possibly non-consecutive – applications or  $i$ AR to all coordinates of the functor, we can omit one  $j$ VR and the ‘innermost’  $i$ AR. [Similarly in the case where translations  $\varphi(\vec{a}, ((s, b^4), d))$  and  $\alpha(\vec{c}, b^4)$  of a functor and (the  $j-1$ -ary main functor of) its  $i$ -th argument contain two consecutive applications of  $i$ AR to all coordinates of the functor which ‘answer’ two – possibly non-consecutive – applications or  $j$ VR to all coordinates of (the main functor of) that argument. There we can skip one  $i$ AR and the ‘outermost’  $j$ VR.]

(XI) If  $\varphi = X_1(\dots(X_k(iAR(X_{k+1}(\dots(X_n(iAR(\{\varphi_1, \dots, \varphi_s\})))\dots))))\dots)$  [ $X_1 \dots X_n = GC, GD, AR$  or  $PVR (p > i)$ ] and  $\alpha = Y_1(\dots(Y_m(jVR(jVR(\{\alpha_1, \dots, \alpha_t\})))\dots))\dots)$  [ $Y_1 \dots Y_m = GD, GC, AR$  or  $qVR (q \geq j)$ ], where  $\#(iAR \text{ in } X_{k+1} \dots X_n) = 0$  and  $\#(iAR \text{ in } X_1 \dots X_k) = \#(jVR + jAR \text{ in } Y_1 \dots Y_m) = r$ , then  $\varphi(\vec{\psi})(\wedge\alpha(\vec{\chi})) \Leftrightarrow \varphi'(\vec{\psi})(\wedge\alpha'(\vec{\chi}))$  for  $\varphi' = X_1(\dots(X_i(iAR(X_{i+1}(\dots(X_n(\{\varphi_1, \dots, \varphi_s\})))\dots))))\dots)$  and  $Y_1(\dots(Y_m(jVR(\{\alpha_1, \dots, \alpha_t\})))\dots) = \alpha'$  [ $\text{length}(\vec{\psi}) = i-1, \text{length}(\vec{\chi}) = j-1$ ].

Proof by induction on  $r$  (analogous to (IX)).

$r=0$ : Since  $\varphi \Leftrightarrow \lambda\vec{x}\lambda v\lambda\vec{y}.\Phi_1[\vee v(\wedge\lambda v'.\Phi_2[\vee v(\wedge\lambda v''.\varphi_1)]), \dots, \vee v(\wedge\lambda v'.\Phi_2[\vee v(\wedge\lambda v''.\varphi_s)])]$ ,

$\varphi' \Leftrightarrow \lambda \vec{x} \lambda v' \lambda \vec{y}. \Phi_1[\forall v'(\wedge \lambda v''. \Phi_2[\varphi_1, \dots, \varphi_s])] \text{ (where } v'' \text{ does not occur in } \Phi_2),$   
 $\alpha \Leftrightarrow \lambda \vec{z} \lambda f. A[\forall f(\wedge \lambda f''. \forall f'(\wedge \lambda f'''. \alpha_1)), \dots, \forall f(\wedge \lambda f''. \forall f'(\wedge \lambda f'''. \alpha_t))], \text{ and}$   
 $\alpha' \Leftrightarrow \lambda \vec{z} \lambda f. A[\forall f'(\wedge \lambda f'''. \alpha_1), \dots, \forall f'(\wedge \lambda f'''. \alpha_t)], \text{ we have } \varphi(\vec{\psi})(\wedge \alpha(\vec{\chi})) \Leftrightarrow \delta =$   
 $\lambda \vec{y}. \Phi_1[A[\Phi_2[\varphi_1 \langle v'' := \wedge \lambda f'''. \alpha_1 \rangle], \dots, \Phi_2[\varphi_1 \langle v'' := \wedge \lambda f'''. \alpha_t \rangle]] \langle \vec{z} := \vec{\chi} \rangle,$   
 $\dots, A[\Phi_2[\varphi_s \langle v'' := \wedge \lambda f'''. \alpha_1 \rangle], \dots, \Phi_2[\varphi_s \langle v'' := \wedge \lambda f'''. \alpha_t \rangle]] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle,$   
 and  $\varphi'(\vec{\psi})(\wedge \alpha'(\vec{\chi})) \Leftrightarrow \varepsilon =$   
 $\lambda \vec{y}. \Phi_1[A[\Phi_2[\varphi_1 \langle v'' := \wedge \lambda f'''. \alpha_1 \rangle], \dots, \Phi_2[\varphi_1 \langle v'' := \wedge \lambda f'''. \alpha_t \rangle]] \langle \vec{z} := \vec{\chi} \rangle,$   
 $\dots, A[\Phi_2[\varphi_s \langle v'' := \wedge \lambda f'''. \alpha_1 \rangle], \dots, \Phi_2[\varphi_s \langle v'' := \wedge \lambda f'''. \alpha_t \rangle]] \langle \vec{z} := \vec{\chi} \rangle \langle \vec{x} := \vec{\psi} \rangle.$

But because  $v''$  does not occur in  $\Phi_2$ :  $\delta \Leftrightarrow \varepsilon$ .

$r > 0$ : Since  $r > 0$ , we can assume that  $\alpha \Leftrightarrow \lambda \vec{z} \lambda w. A[\forall w(\wedge \alpha_1(\vec{z}')), \dots, \forall w(\wedge \alpha_w(\vec{z}'))]$  and  
 $\varphi \Leftrightarrow \lambda \vec{x} \lambda v \lambda \vec{y}. \Phi[\forall v(\wedge \lambda v'. \varphi_1(\vec{x}')(v')(\vec{y}')), \dots, \forall v(\wedge \lambda v'. \varphi_v(\vec{x}')(v')(\vec{y}'))],$  for  $v \leq s$ ,  $w \leq t$ ,  
 $\text{length}(\vec{x}') = i-1$ , and  $\text{length}(\vec{z}') = j-1$ . Note that  $\varphi_1, \dots, \varphi_v$  and  $\alpha_1, \dots, \alpha_w$  are translations  
 that have undergone  $r$  times  ${}^i\text{AR}$  and  ${}^j\text{VR}$ , respectively. So, we can apply the induction  
 hypothesis: there are simpler terms  $\varphi_1', \dots, \varphi_v'$  and  $\alpha_1', \dots, \alpha_w'$  (with  $r-1$  times  ${}^i\text{AR}$  and  
 ${}^j\text{VR}$ ) such that for all  $i$  and  $j$ ,  $1 \leq i \leq v$ ,  $1 \leq j \leq w$ :  $\varphi_i(\vec{x}')(\wedge \alpha_j(\vec{z}')) \Leftrightarrow \varphi_i'(\vec{x}')(\wedge \alpha_j'(\vec{z}'))$ .  
 But then for  $\varphi'' = \lambda \vec{x} \lambda v \lambda \vec{y}. \Phi[\forall v(\wedge \lambda v'. \varphi_1'(\vec{x}')(v')(\vec{y}')), \dots, \forall v(\wedge \lambda v'. \varphi_v'(\vec{x}')(v')(\vec{y}'))],$   $\alpha''$   
 $= \lambda \vec{z} \lambda w. A[\forall w(\wedge \alpha_1'(\vec{z}')), \dots, \forall w(\wedge \alpha_w'(\vec{z}'))]$  we have  $\varphi(\vec{\psi})(\wedge \alpha(\vec{\chi})) \Leftrightarrow \varphi''(\vec{\psi})(\wedge \alpha''(\vec{\chi})).$

These facts suffice to check the claims about numbers of readings for the sentences  
 treated in the paper. They also suggest minimal ways of obtaining these readings: (i)  
 apply AL only to basic lexical translations; (ii) apply VR and AR only to translations of  
 lexical items and coordinations; (iii) apply only  ${}^1\text{VR}$  to S, T, and CN;  ${}^1\text{AR}$ ,  ${}^2\text{AR}$  and  
 ${}^3\text{VR}$  to TV and PV; and  ${}^1\text{AR}$  and  ${}^2\text{VR}$  to IV and Det; (iv) apply no VR to (coordinates of)  
 the main functor of the sentence; (v) do no final applications of  ${}^j\text{VR}$  to (the  $j-1$ -ary main  
 functor of) an  $i$ -th argument which are ‘answered’ by an application of  ${}^i\text{AR}$  to (the  
 coordinates of) its functor; (vi) do no consecutive applications of  ${}^j\text{VR}$  to (the coordinates  
 of) (the  $j-1$ -ary main functor of) an  $i$ -th argument which are ‘answered’ by two  
 applications of  ${}^i\text{AR}$  to (the coordinates of) its functor; (vi) do no consecutive applications  
 of  ${}^i\text{AR}$  to (the coordinates of) a functor which are ‘answered’ by two applications of  ${}^j\text{VR}$   
 to (the coordinates of) (the  $j-1$ -ary main functor of) its  $i$ -th argument. Two examples:

- (9) Mary walks and seeks every unicorn  
 (126) Fred claims that every schoolboy believes that a mathematician wrote *Through the  
 looking glass*.

If we assume (for simplicity) that *every unicorn* is a lexical item, we see that there are five  
 candidates for type change: the four lexical items and the coordination *walks and seeks  
 every unicorn*. *Seeks* can undergo  ${}^1\text{AL}$ . The proper name only gives rise to final  ${}^1\text{VR}$   
 applications which have to be answered by  ${}^2\text{AR}$  (on both conjoined verb phrases or on  
 their conjunction), so no type change is needed here (v). The quantifier *every unicorn*  
 necessitates  ${}^1\text{AR}$  for lowered *seeks*.  ${}^1\text{VR}$  for this noun phrase would be ‘final’ and  
 ‘answered’ by  ${}^1\text{AR}$  on *seeks*, hence the verb needs at most one  ${}^1\text{AR}$ . For the conjoined

verb phrases and their conjunction, VR is ruled out (iv). We saw that  ${}^2\text{AR}$  on the conjoined verb phrases or their conjunction would have to be answered by a final application of  ${}^1\text{VR}$  to the proper name. So the choice is between  $\langle \rangle$  (nothing) and  $\langle {}^1\text{AL}, {}^1\text{AR} \rangle$  for *seeks*: two readings.

Sentence (126) does not give rise to argument lowerings; there are no coordinations, so the type change can be kept lexical. As for the term phrases, we see that the proper names *Fred* and *Through the looking glass* only give rise to final  ${}^1\text{VR}$  applications which have to be answered by their functors *claims* and *wrote* – no type change is needed here (v). However, the quantified noun phrases *every schoolboy* and *a mathematician* necessitate (‘unanswered’) applications of  ${}^2\text{AR}$  in their functors *believes* and *wrote*, respectively. Iteration of  ${}^2\text{AR}$  on these functors would involve applications of final  ${}^1\text{VR}$  to the quantifiers (v). For *wrote*, then, we have exactly one  ${}^2\text{AR}$  and zero  ${}^1\text{AR}$ . The remaining option,  ${}^3\text{VR}$ , is necessarily answered; hence it must be non-final (v) and non-consecutive (vi). So, we are left with  $(\alpha) \langle {}^2\text{AR} \rangle$  and  $(\beta) \langle {}^3\text{VR}, {}^2\text{AR} \rangle$ . For *claims*, we have zero  ${}^2\text{AR}$ . Besides,  ${}^3\text{VR}$  is impossible for this main functor of the sentence: value raising would not lead to a translation of type t (iv).  ${}^1\text{AR}$  will be answered; hence it must be non-consecutive (vii). I.e., (a)  $\langle \rangle$  and (b)  $\langle {}^1\text{AR} \rangle$  remain. For *believes*, we have exactly one  ${}^2\text{AR}$  and two possibilities,  ${}^1\text{AR}$  and  ${}^3\text{VR}$ . The facts about *wrote* entail that  ${}^1\text{AR}$  will be applied at most once.  ${}^3\text{VR}$  is necessarily answered; hence it must be non-final. Moreover, we saw that applying it more than once would violate (vii). Therefore, the remaining options are (A)  $\langle {}^2\text{AR} \rangle$ , (B)  $\langle {}^1\text{AR}, {}^2\text{AR} \rangle$ , (C)  $\langle {}^2\text{AR}, {}^1\text{AR} \rangle$ , (D)  $\langle {}^3\text{VR}, {}^2\text{AR} \rangle$ , (E)  $\langle {}^2\text{AR}, {}^3\text{VR}, {}^1\text{AR} \rangle$ , (F)  $\langle {}^1\text{AR}, {}^3\text{VR}, {}^2\text{AR} \rangle$ , (G)  $\langle {}^3\text{VR}, {}^2\text{AR}, {}^1\text{AR} \rangle$  and (H)  $\langle {}^3\text{VR}, {}^1\text{AR}, {}^2\text{AR} \rangle$ . Only the combinations (A)(a)( $\alpha$ ), (B)(a)( $\beta$ ), (C)(a)( $\beta$ ), (D)(b)( $\alpha$ ), (E)(b)( $\beta$ ), (F)(b)( $\beta$ ), (G)(b)( $\beta$ ) and (H)(b)( $\beta$ ) fit: eight readings.

## REFERENCES

- Asher, N., and D. Bonevac (1985). 'How Extensional is Extensional Perception?'. *Linguistics and Philosophy* 8, 203–228.
- Asher, N., and D. Bonevac (1987). 'Determiners and Resource Situations'. *Linguistics and Philosophy* 10, 567–596.
- Bach, E. (1980). 'Tenses and Aspects as Functions on Verb-phrases'. In C. Rohrer (ed.), *Time, Tense and Quantifiers*. Tübingen: Max Niemeyer.
- Barwise, J. (1981). 'Scenes and Other Situations'. *Journal of Philosophy* 78, 369–397.
- Benthem, J. van (1986). *Essays on Logical Semantics*. Dordrecht: Reidel.
- Benthem, J. van (1988). 'The Lambek Calculus'. In R. Oehrle, E. Bach and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*. Dordrecht: Reidel.
- Bouma, G. (1986). 'Lambek Semantics'. Manuscript.
- Cooper, R. (1983). *Quantification and Syntactic Theory*. Dordrecht: Reidel.
- Dekker, P. (1988). 'Flexible Flexibilities. Notes on Coordination and Quantification'. Manuscript, University of Amsterdam.
- Does, J. van der (1990). 'A Generalized Quantifier Logic for Naked Infinitives'. DYANA Deliverable R.2.1.A.
- Dowty, D. (1988). 'Type Raising, Functional Composition, and Non-Constituent Conjunction'. In R. Oehrle, E. Bach and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*. Dordrecht: Reidel.
- Dowty, D., R. Wall, and S. Peters (1981). *Introduction to Montague Semantics*. Dordrecht: Reidel.
- Dummett, M. (1976). *Frege. Philosophy of Language*. London: Duckworth.
- Emms, M. (1989). 'Polymorphic Quantifiers'. Paper presented at the 7th Amsterdam Colloquium.
- Frege, G. (1884). *Die Grundlagen der Arithmetik. Eine logisch-mathematische Untersuchung über den Begriff der Zahl*. Breslau: Köbner. Reprint (1961) published by Georg Olms, Hildesheim.
- Gamut, L. (1982). *Logica, taal en betekenis 2: intensionele logica en logische grammatica*. Utrecht/Antwerpen: Het Spectrum.
- Gazdar, G. (1980). 'A Cross-categorial Semantics for Coordination'. *Linguistics and Philosophy* 3, 407–410.
- Gazdar, G., E. Klein, G. Pullum, and I. Sag (1985). *Generalized Phrase Structure Grammar*. Oxford: Basil Blackwell.
- Groenendijk, J., and M. Stokhof (1984). *Studies on the Semantics of Questions and the Pragmatics of Answers*. Dissertation University of Amsterdam.
- Groenendijk, J., and M. Stokhof (1987). 'Type-shifting Principles and the Semantics of Interrogatives'. University of Amsterdam: ITLI Prepublication Series 87-01.
- Hendriks, H. (1988). 'Type Change in Semantics: the Scope of Quantification and Coordination'. In E. Klein and J. van Benthem (eds.), *Categories, Polymorphism and Unification*, University of Edinburgh: Centre for Cognitive Science, and Amsterdam: Institute for Language, Logic and Information.
- Hendriks, H. (1989). 'Cut Elimination and Semantics in Lambek Calculus'. Manuscript. University of Amsterdam.
- Janssen, T. (1983). *Foundations and Applications of Montague Grammar*. Dissertation University of Amsterdam.
- Janssen, T. (1984). 'Individual Concepts are Useful'. In F. Landman and F. Veltman (eds.), *Varieties of Formal Semantics*, Dordrecht: Foris.
- Kamp, H. (1984). 'A Theory of Truth and Semantic Representation'. In J. Groenendijk, T. Janssen and M. Stokhof (eds.), *Truth, Interpretation and Information*. Dordrecht: Foris.
- Keenan, E., and L. Faltz (1985). *Boolean Semantics for Natural Language*. Dordrecht: Reidel.

- Klein, E., and I. Sag (1985). 'Type-driven Translation'. *Linguistics and Philosophy* 8, 163-202.
- Lambek, J. (1958). 'The Mathematics of Sentence Structure'. *American Mathematical Monthly* 65, 154-169. Reprinted in W. Buszkowski, W. Marciszewski and J. van Benthem (eds.) (1989), *Categorial Grammar*. Amsterdam: Benjamins.
- Landman, F., and I. Moerdijk (1983). 'Compositionality and the Analysis of Anaphora'. *Linguistics and Philosophy* 6, 89-114.
- Montague, R. (1974a). 'Universal Grammar'. In R. Thomason (ed.), *Formal Philosophy*. New Haven: Yale University Press.
- Montague, R. (1974b). 'The Proper Treatment of Quantification in Ordinary English'. In R. Thomason (ed.), *Formal Philosophy*. New Haven: Yale University Press.
- Moortgat, M. (1988). *Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calculus*. Dordrecht: Foris.
- Moortgat, M. (1990). 'The Quantification Calculus: Questions of Axiomatization'. DYANA Deliverable R1.2.A.
- Muskens, R. (1989). *Meaning and Partiality*. Dissertation University of Amsterdam.
- Partee, B. (1986). 'Noun Phrase Interpretation and Type-shifting Principles'. In J. Groenendijk, D. de Jongh and M. Stokhof (eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*. Dordrecht: Foris.
- Partee, B., and M. Rooth (1983). 'Generalized Conjunction and Type Ambiguity'. In R. Bäuerle, C. Schwarze and A. von Stechow (eds.), *Meaning, Use and Interpretation of Language*. Berlin: de Gruyter.
- Pollard, C., and I. Sag (1983). 'Reflexives and Reciprocals in English: an Alternative to the Binding Theory'. In M. Barlow, D. Flickinger and M. Wescot (eds.), *Proceedings of the Second West Coast Conference on Formal Linguistics*. Stanford: Stanford Linguistics Department.
- Reinhart, T. (1983). 'Coreference and Bound Anaphora: a Restatement of the Anaphora Questions'. *Linguistics and Philosophy* 6, 47-88.
- Rooth, M., and B. Partee (1982). 'Conjunction, Type Ambiguity, and Wide Sope "Or"'. In D. Flickinger, M. Macken and N. Wiegand, *Proceedings of the 1982 West Coast Conference on Formal Linguistics*. Stanford: Stanford Linguistics Department.

*Department of Philosophy  
University of Amsterdam  
Nieuwe Doelenstraat 15  
1012 CP Amsterdam  
The Netherlands*



# The ITLI Prepublication Series

1990

*Logic, Semantics and Philosophy of Language*

LP-90-01 Jaap van der Does  
LP-90-02 Jeroen Groenendijk, Martin Stokhof  
LP-90-03 Renate Bartsch  
LP-90-04 Aarne Ranta  
LP-90-05 Patrick Blackburn  
LP-90-06 Gennaro Chierchia  
LP-90-07 Gennaro Chierchia  
LP-90-08 Herman Hendriks  
LP-90-09 Paul Dekker

LP-90-10 Theo M.V. Janssen  
*Mathematical Logic and Foundations*

ML-90-01 Harold Schellinx  
ML-90-02 Jaap van Oosten  
ML-90-03 Yde Venema  
ML-90-04 Maarten de Rijke  
ML-90-05 Domenico Zambella

*Computation and Complexity Theory*

CT-90-01 John Tromp, Peter van Emde Boas  
CT-90-02 Sieger van Denneheuvel

Gerard R. Renardel de Lavalette  
CT-90-03 Ricard Gavaldà, Leen Torenvliet  
Osamu Watanabe, José L. Balcázar

CT-90-04 Harry Buhrman, Leen Torenvliet

*Other Prepublications*

X-90-01 A.S. Troelstra

X-90-02 Maarten de Rijke

X-90-03 L.D. Beklemishev

X-90-04

X-90-05 Valentin Shehtman

X-90-06 Valentin Goranko, Solomon Passy

X-90-07 V.Yu. Shavrukov

X-90-08 L.D. Beklemishev

X-90-09 V.Yu. Shavrukov

X-90-10 Sieger van Denneheuvel

Peter van Emde Boas

X-90-11 Alessandra Carbone

A Generalized Quantifier Logic for Naked Infinitives  
Dynamic Montague Grammar  
Concept Formation and Concept Composition  
Intuitionistic Categorical Grammar  
Nominal Tense Logic  
The Variability of Impersonal Subjects  
Anaphora and Dynamic Logic  
Flexible Montague Grammar  
The Scope of Negation in Discourse,  
towards a flexible dynamic Montague grammar  
Models for Discourse Markers

Isomorphisms and Non-Isomorphisms of Graph Models  
A Semantical Proof of De Jongh's Theorem  
Relational Games  
Unary Interpretability Logic  
Sequences with Simple Initial Segments

Associative Storage Modification Machines  
A Normal Form for PCSJ Expressions

Generalized Kolmogorov Complexity  
in Relativized Separations  
Bounded Reductions

Remarks on Intuitionism and the Philosophy of Mathematics,  
Revised Version  
Some Chapters on Interpretability Logic  
On the Complexity of Arithmetical Interpretations of Modal Formulae  
Annual Report 1989  
Derived Sets in Euclidean Spaces and Modal Logic  
Using the Universal Modality: Gains and Questions  
The Lindenbaum Fixed Point Algebra is Undecidable  
Provability Logics for Natural Turing Progressions of Arithmetical  
Theories  
On Rosser's Provability Predicate  
An Overview of the Rule Language RL/1  
Provable Fixed points in  $\text{ID}_0 + \Omega_1$ , revised version