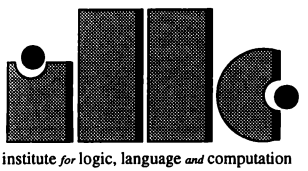
Studied Flexibility

ILLC Dissertation Series 1993–5



For further information about ILLC-publications, please contact:

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-5256090
fax: +31-20-5255101
e-mail: illc@fwi.uva.nl

Studied Flexibility

Categories and Types in Syntax and Semantics

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
Prof. Dr. P.W.M. de Meijer
in het openbaar te verdedigen in de Aula der Universiteit
(Oude Lutherse Kerk, ingang Singel 411, hoek Spui),
op woensdag 8 december 1993 te 12.00 uur

door

Hermanus Lambertus Wilhelmus Hendriks
geboren te Nijmegen

Promotor: Prof. Dr. R.I. Bartsch
Faculteit der Wijsbegeerte
Co-promotor: Dr. M.J.B. Stokhof
Faculteit der Wijsbegeerte

Copyright © 1993 Herman Hendriks, Amsterdam
Typeset in LaTeX by the author
Printed by Febodruk, Enschede
ISBN 90–74795–01–3

De chrysanten,
die in de vaas op de tafel
bij het raam staan: dat

zijn niet de chrysanten
die bij het raam
op de tafel
in de vaas staan.

De wind die je zo hindert
en je haar door de war maakt,

dat is de wind die je haar verwart;
het is de wind waardoor je niet
meer gehinderd wilt worden
als je haar in de war is.

Hans Faverey, *Chrysanten, roeiers*

The chrysanthemums
that stand in the vase on the table
by the window: these

are not the chrysanthemums
that stand by the window
on the table
in the vase.

The wind that annoys you
and ruffles your hair

is the wind that ruffles your hair;
it is the wind that you don't any
longer want to be annoyed by
when your hair is ruffled.

Translated by J.M. Coetzee

Preface

> ... because if I'd a knowed what trouble it was
> to make a book I wouldn't a tackled it
> and ain't a-going to no more.
>
> Mark Twain, *The Adventures of Huckleberry Finn*

First of all, I would like to thank Caroline Aartsen, Paul Dekker and Martin Stokhof, without whom you would have had to write the rest of this book yourself, as well as Renate Bartsch, Johan van Benthem, Jeroen Groenendijk, Theo Janssen, Michael Moortgat, Dirk Roorda and Frank Veltman, without whom I would have had to write the rest of this book myself. I am also grateful for the financial support of the Netherlands Organization for Scientific Research (NWO), project 220-192-200, and ESPRIT Basic Research Actions 3175 and 6852, *Dynamic Interpretation of Natural Language* (DYANA). And I am pleased to express my appreciation of the interest, help, encouragement and criticism I received from friends and relatives, colleagues and teachers, students and more or less anonymous referees, of whom I would like to reveal Erik Aarts, Elfriede Aartsen, Irene Aartsen, Piet Aartsen, Emmon Bach, Guy Barry, David Beaver, André Beening, Ria Beentjes, Martin van den Berg, Hans den Besten, Inge Bethke, Peter Blok, Theodoor Bolten, Gosse Bouma, Hans Broekhuis, David Bromberg, Woichiech Buszkowski, Bob Carpenter, Gennaro Chierchia, Saskia Daalder, Kees van Deemter, Stef van Dijk, Jaap van der Does, Marjon van der Does, Jeroen van Dongen, Tineke van Dongen, Jan van Eijck, Martin Emms, Maarten Franssen, Anna Gavarró, Gerald Gazdar, Edwin Gerdsen, Willem Groeneveld, Annemieke de Groot, Wiltea de Groot, Ton van Haaften, Christa de Haan, Ken Hale, Fritz Hamm, Ad Heemskerk, Berrie Heesen, Bert Hendriks, Dory Hendriks, Ineke Hendriks, Kees Hendriks, Margreet Hendriks, Petra Hendriks, Rini Hendriks, Mark Hepple, Dirk Heylen, Jack Hoeksema, Pauline Jacobson, José Janssen, Dick de Jongh, Marc Jumelet, Hans Kamp, Edward Keenan, Ewan Klein, Wim Klooster, Netteke Koene, Tiina Komi, Marcus Kracht, Emiel Krahmer, Manfred Krifka, Fred Landman, Marjan Landman,

Serge Lapierre, Alain Lecomte, Karel van der Leeuw, Jacques van Leeuwen, Michiel Leezenberg, Noor van Leusen, Lucas Ligtenberg, Erik-Jan van der Linden, Christel Loeffen, Karin Loeffen, Rob Loeffen, Stefan Loeffen, Sietze Looyenga, Peter Luiken, Jaap Maat, Frits van der Meij, Anne-Marie Mineur, Grisha Mints, Claire Moll, Michael Morreau, Glyn Morrill, Reinhard Muskens, Arthur Nieuwendijk, Richard Oehrle, Leonoor Oversteegen, Barbara Partee, Jacques Penders, Marjorie Pigge, Ans Post, Andreja Prijatelj, Hub Prüst, Stephan Raaijmakers, Aarne Ranta, Mike Reape, Tanya Reinhart, Rob Remmen, Ans Remmits, Peter Rijpkema, Cees Roele, Lucia Ruehling, Peter Ruhrberg, Rob van der Sandt, Remko Scha, André Schenk, Maarten Schoonwater, Jerry Seligman, Pieter Seuren, Jacomina Snijders, Inge Sol, Mark Steedman, Harry Stein, George Steiner, Abram de Swaan, Anna Szabolcsi, Yao-Hua Tan, Leen Torenvliet, Hans Uszkoreit, Yde Venema, Bruno Verbeek, Henk Verkuyl, Kees Vermeulen, Stephan Vermeulen, Koen Versmissen, Roel de Vrijer, Cornelia Vrolijk, Heinrich Wansing, Reni Webb, Henk Westenend, Ton van der Wouden, Henk Zeevat, Ede Zimmerman and Frans Zwarts.

Table of Contents

Introduction: Categories and Types 11
1 Types 12
2 Categories 16
3 Types and Categories 19

Chapter One: Flexible Montague Grammar 27
1 Syntax and Semantics 28
1.1 Categories 29
1.2 Types 30
1.3 Translation Rules 32
1.3.1 Lexical Expressions 32
1.3.2 Compound Expressions 34
1.4 Scope Ambiguities: Quantifying-in 35
1.5 Scope Ambiguities: Quantifier Storage 37
2 Arguments for Flexibility 43
3 Flexibility in Syntax and Semantics 57
3.1 Scope Ambiguities: L 61
3.2 Scope Ambiguities: LP 68
4 The Fragment 70
4.1 Basic Translations 72
4.2 Derived Translations 75
5 Applications 79
5.1 The Scope of Quantification 81
5.1.1 Flat Structures 81
5.1.2 Embedded Structures 85
5.2 The Scope of Coordination 89
5.3 Structural Restrictions 95
5.3.1 Scope Islands 95
5.3.2 Scope Sieves 108
6 Flexibility and Anaphora 110
Appendix: Semantic Properties of the Fragment 118

Chapter Two: Compositionality and Flexibility 129
1 Interpretation and Translation 129
2 Compositionality 135
3 Similarity 145
3.1 AddOp 153
3.2 AddSorts 158
3.3 DelOp 160
3.4 DelSorts 162
4 Interpretability 164
5 Flexible Interpretation 177

Chapter Three: Lambek Semantics 187
1 Introduction 188
2 Cut Elimination and Semantics 194
3 Further Normalization 200
4 Spurious Ambiguity 212
5 Partial Deduction and Proof Nets 219
5.1 Partial Deduction 219
5.1 Proof Nets 231
6 Encoding Atomic Categories 246

References 257

Summary in Dutch 267

Introduction

Categories and Types

I'm writing a book. That's all – just a book.
But if I don't take it seriously, how can I hope to get it done?

Paul Auster, *The New York Trilogy*

In theories of formal grammar it has become customary to assume that linguistic expressions belong to syntactic *categories*, whereas their interpretations inhabit semantic *types*.

The present dissertation is an exploration of the consequences of this twofold assumption, and this introduction aims to set out the basic ideas of logical syntax and semantics as they are found in categorial grammar and lambda calculus, to focus on their convergence in theories of linguistic syntax and semantics, and to situate the contributions of the present dissertation within that convergence.

The idea that the objects of thought form a hierarchy based on function-argument relationships traces back to the mathematical and philosophical works of Gottlob Frege, which gave rise to two streams of logical investigation. On the one hand, Bertrand Russell's *theory of types* initiated a type-theoretic tradition in the foundations of mathematics which generated important research programs such as the lambda calculus. On the other, Lesniewski and Ajdukiewicz developed a technical theory of categories which eventually resulted in the linguistic paradigm that has become known as *categorial grammar*. A first convergence of both streams of research may be observed in the seminal logico-linguistic papers of Montague (1970a, 1970b, 1973), where mathematically inspired type theories supply the semantics for a categorially informed syntax of natural language fragments. Montague's view on the relationship between the syntactic and the semantic component has been very influential. Among its numerous offspring, recent work building on Lambek's (1958) extension of classical categorial grammar distinguishes itself by assuming an even tighter connection between the two components.

1 Types

In 1902, Russell wrote a letter to Frege[1] in which he showed that a naive principle of set comprehension leads to a paradox for the set $A = \{x \mid x \notin x\}$: $A \in A$ if and only if $A \notin A$. Russell solved the problem by simply excluding $x \in x$ from the set of well-formed formulas. Roughly, every variable $x, y, z, \ldots$ is annotated (or *typed*) with an index $n \in \{1, 2, \ldots\}$, and $x_n \in y_m$ is well-formed if and only if $m = n + 1$.

Within formal set theory, this theory of types was superseded by Zermelo's axiomatic approach, in which the axiom of foundation requires that each set be well-founded, but the idea of typing recurred in the context of the lambda calculus. The lambda calculus was originally conceived by Church (1932/3) as part of a general theory about functions and logic, and intended as a foundation for mathematics. Although the full system turned out to be inconsistent, the subsystem dealing with functions only turned out to be a successful model for computable functions. This subsystem, the pure, type-free lambda calculus, can be seen as a theory about functions as *rules* (prescriptions).[2]

Typed lambda calculus, which we will be concerned with, can be regarded as a theory about functions as *graphs* (sets of ordered pairs).[3] In the (Church-style) typed lambda calculus we consider, every term has a unique type: it is an annotated version of a type-free lambda term. Within this theory, all the usual logical operators (truth, falsity, Boolean connectives and quantifiers) are definable in terms of application, abstraction and identity.[4]

The set T of *types* consists of primitive types and types formed out of them by means of certain operations. Common primitive types in logical semantics include t (the type of *truth values*) and e (the type of *entities*). We will only consider the type-forming operator that forms so-called *function* types (σ, τ) out of types σ and τ.

A *frame* $F = (D_{E,\tau})_{\tau \in T}$ is based on some non-empty set E, the *domain*

[1] See Van Heijenoort (ed.) (1967).

[2] A fine survey of the type-free lambda calculus is given in Barendregt (1981).

[3] See Volume II of Gamut (1991) for a more enjoyable introduction to the typed lambda calculus.

[4] See Henkin (1963) and Gallin (1975). Observe that $\top$ (truth) can be defined as $[\lambda v_{0,t}\, v_{0,t} = \lambda v_{0,t}\, v_{0,t}]$; $\bot$ (falsity) as $[\lambda v_{0,t}\, v_{0,t} = \lambda v_{0,t}\, \top]$; $\neg\varphi$ (negation) as $[\varphi = \bot]$; $[\varphi \wedge \psi]$ (conjunction) as $[\lambda v_{0,(t,t)}\, [[[v_{0,(t,t)}](\varphi)] = \psi] = \lambda v_{0,(t,t)}\, [[v_{0,(t,t)}](\top)]]$; and $\forall v_{i,\tau}\, \varphi$ (universal quantification) as $[\lambda v_{i,\tau}\, \varphi = \lambda v_{i,\tau}\, \top]$. Furthermore, $[\varphi \vee \psi]$ (disjunction) and $[\varphi \rightarrow \psi]$ (implication) are definable as $\neg[\varphi \wedge \psi]$ and $[\neg\varphi \vee \psi]$, respectively; and $\exists v_{i,\tau}\varphi$ (existential quantification) as $\neg\forall v_{i,\tau}\neg\varphi$.

of individuals or *entities*, and it consists of a family of domains $D_{E,\tau}$, one for each type τ, such that

- $D_{E,t} = \{1, 0\}$, the set of truth values;[5]
- $D_{E,e} = E$; and
- $D_{E,(\sigma,\tau)} = D_{E,\tau}^{D_{E,\sigma}}$, the set of functions from objects in $D_{E,\sigma}$ to objects in $D_{E,\tau}$.[6]

The formal language to refer to objects in such structures consists of *terms*. There are simple terms and compound terms. The simple terms include an infinite number of variables $v_{i,\tau}$ ($v_{i,\tau}$ is a variable of type τ for each $i \in \mathbb{N}$), and (possibly) typed constants c_τ. The compound terms are constructed as follows: if α and β are terms of type (σ, τ) and type τ, respectively, then $[[\alpha](\beta)]$ is a term of type τ (*application*); if α is a term of type τ and $v_{i,\sigma}$ is a variable, then $\lambda v_{i,\sigma}\, \alpha$ is a term of type (σ, τ) (*abstraction*); and if α and β are terms of type τ, then $[\alpha = \beta]$ is a term of type t (*identity*).[7]

Next, let an *interpretation* of the constants be a function I having the set of all constants as its domain such that if c is a constant of type τ, then $I(c) \in D_{E,\tau}$, and let a *variable assignment* a be a function having the set of all variables as its domain such that for all $v_{i,\tau}$: $a(v_{i,\tau}) \in D_{E,\tau}$. We write $a[v_{i,\tau}/d]$ for the assignment a' such that (*i*) $a'(v_{i,\tau}) = d$ and (*ii*) $a'(v_{i',\tau'}) = a(v_{i',\tau'})$ if $i' \neq i$ or $\tau' \neq \tau$.

A *model* m is an ordered pair consisting of a frame and an interpretation of the constants:

(1) $\quad m = \langle (D_{E,\tau})_{\tau \in T}, I \rangle$

Now, $|\alpha|^{m,a}$, the *interpretation* (or *extension*) of a term α in a model m under a variable assignment a is defined as follows:

- $|c_\tau|^{m,a} = I(c_\tau)$ for constants c_τ;
- $|v_{i,\tau}|^{m,a} = a(v_{i,\tau})$ for variables $v_{i,\tau}$;

[5] Where 1 and 0 represent 'true' and 'false', respectively.

[6] By performing 'Schönfinkel's trick' (1924), we can obtain functions of several arguments by iteration of application.

[7] If no confusion is likely to arise, we will write terms $[[\alpha](\beta)]$ as $[\alpha(\beta)]$ or as $\alpha(\beta)$. Whenever the latter abbreviation is used repeatedly, the association is to the left: $\alpha(\beta_1)\ldots(\beta_n)$ has to be read as $[[\ldots[[\alpha](\beta_1)]\ldots](\beta_n)]$. On top of this, we will employ $\lambda v_{i_1,\tau_1}\ldots\lambda v_{i_m,\tau_m}.\,\alpha(\beta_1)\ldots(\beta_n)$ as shorthand for $\lambda v_{i_1,\tau_1}\ldots\lambda v_{i_m,\tau_m}[[\ldots[[\alpha](\beta_1)]\ldots](\beta_n)]$.

- $|[[\alpha](\beta)]|^{m,a} = |\alpha|^{m,a}(|\beta|^{m,a})$;
- $|\lambda v_{i,\sigma}\, \alpha|^{m,a} = f \in D_{E,\tau}^{D_{E,\sigma}}$ such that for all d: $f(d) = |\alpha|^{m,a[v_{i,\sigma}/d]}$; and
- $|[\alpha = \beta]|^{m,a} = 1$ iff $|\alpha|^{m,a} = |\beta|^{m,a}$.

By way of example, let $P_{(e,t)}$ and x_e denote some variables $v_{i,(e,t)}$ and $v_{j,e}$ which (due to their type) range over (characteristic functions of) sets of individuals,[8] and entities, respectively, and let PC and MALFUNCTION denote the constants $\text{PC}_{(e,t)}$ and $\text{MALFUNCTION}_{(e,t)}$ which are interpreted as (the characteristic functions of) the set of the entities that are personal computers, and the set of malfunctioning entities, respectively.

Then $\lambda P_{(e,t)} \exists x_e[\text{PC}(x_e) \wedge P_{(e,t)}(x_e)]$[9] is a complex term of type $((e,t),t)$ which denotes the function from sets of entities to truth values such that a set is assigned the value 1 iff there is a personal computer in that set: a plausible interpretation for the noun phrase *a personal computer*.

Next, consider the term $[\lambda P_{(e,t)} \exists x_e[\text{PC}(x_e) \wedge P_{(e,t)}(x_e)]](\text{MALFUNCTION})$ of type t. The value of this term is the above function applied to the set of malfunctioning entities. It will be 1 ('true') iff there is a personal computer which malfunctions. And in fact, the latter term turns out to be equivalent to the term $\exists x[\text{PC}(x) \wedge \text{MALFUNCTION}(x)]$ of type t.[10]

This is an instance of a general equivalence: $[\lambda x\, \alpha](\beta) \Leftrightarrow \alpha[x{:=}\beta]$. Here, $\alpha[x{:=}\beta]$ denotes the result of substituting β for all free occurrences of x in α, and the condition is that the substitution causes no free variable in β to become bound in the result $\alpha[x{:=}\beta]$. The reduction of $[\lambda x\, \alpha](\beta)$ to $\alpha[x{:=}\beta]$ is called *λ-conversion* (or *β-conversion*).[11]

There are various alternative ways of setting up the typed lambda calculus. For example, Montague's **IL** ('Intensional Logic' (1974)),[12] the calculus which will be employed in Chapter 1, is a version of type theory which enables one to form *intensional* types (s,τ) out of types τ. Its frames include

[8] A function f from entities to truth values is the *characteristic function of* a set A of entities iff $f(d) = 1$ just in case $d \in A$.

[9] So, officially (cf. footnote 7): $\lambda v_{i,(e,t)} \exists v_{j,e}[[[\text{PC}_{(e,t)}](v_{j,e})] \wedge [[v_{i,(e,t)}](v_{j,e})]]$.

[10] Terms α and β are *equivalent* ($\alpha \Leftrightarrow \beta$) iff for all m, a: $|\alpha|^{m,a} = |\beta|^{m,a}$.

[11] Other familiar equivalences in the typed lambda calculus are known as *α-conversion* (or *renaming bound variables*): $\lambda x\, \alpha \Leftrightarrow \lambda y\, [\alpha[x{:=}y]]$, provided that the substitution does not cause occurrences of y to become bound in $\alpha[x{:=}y]$; and *η-conversion*: $\lambda x\, [[\alpha](x)] \Leftrightarrow \alpha$ whenever x does not occur freely in α.

[12] For **IL**, see also Chapter 5 of Volume II of Gamut (1991).

an additional non-empty set W of *possible worlds*, and the definition of domains $D_{E,W,\tau}$ is extended with the provision that $D_{E,W,(s,\tau)} = D^W_{E,W,\tau}$, i.e., the set of functions from possible worlds to objects in $D_{E,W,\tau}$.

The formal language and its interpretation are adapted accordingly. In **IL**, an interpretation of the constants I is a function such that $I(c_\tau) \in D^W_{E,W,\tau}$ for each constant c of type τ. In general, the extension of an **IL**-term α is determined given a model m, a possible world w, and an assignment a, and the extension $|c|^{m,w,a}$ of constants c is defined as $I(c)(w)$. Hence the extension of constants can vary from world to world.

Moreover, in addition to the variables and complex terms that were introduced earlier (and of which the **IL**-extension is defined in essentially the same way – just replace 'm, a' ('$m, a[v_{i,\sigma}/d]$') by 'm, w, a' ('$m, w, a[v_{i,\sigma}/d]$')), the syntax of **IL** permits us to form terms $^\vee\alpha$ of type τ out of terms α of type (s,τ), as well as terms $^\wedge\alpha$ of type (s,τ) out of terms α of type τ. Their extension is defined as follows:

- $|^\vee\alpha|^{m,w,a} = |\alpha|^{m,w,a}(w)$; and
- $|^\wedge\alpha|^{m,w,a} = f \in D^W_{E,W,\tau}$ such that for all $w' \in W$: $f(w') = |\alpha|^{m,w',a}$.

An important additional[13] equivalence in **IL** is $^\vee{}^\wedge$*-elimination*: for all terms α it holds that $^\vee{}^\wedge\alpha \Leftrightarrow \alpha$.[14]

This version of typed lambda calculus allows a formal reconstruction of central semantic notions such as properties, propositions and individual concepts. Gallin (1975) has shown that **IL** can be embedded into the (extensional) type theory outlined above if we enrich it with a basic type s such that $D_{E,W,s} = W$. The resulting system is called 'two-sorted type theory', since it contains two basic types (apart from t).

It is also possible to assign an alternative interpretation to extensional typed lambda calculus. Thus, Orey (1959) has proposed to interpret the calculus in hierarchies of *relations*, rather than functions. Muskens (1989) argues that this relational interpretation is more convenient if we want to

[13] It is worth noting that λ-conversion in **IL** is subject to stronger conditions than in extensional type theory: $[\lambda x\, \alpha](\beta) \Leftrightarrow \alpha[x{:=}\beta]$ if the substitution causes no free variable in β to become bound in the result $\alpha[x{:=}\beta]$ *and* β *is intensionally closed.* The precise definition of the latter notion need not concern us here. It suffices to know that variables and all terms of the form $^\wedge\alpha$ are intensionally closed. (See Section 5.5, 5.6 and 5.8 of Gamut (1991), Volume II, and the discussion in Muskens (1989), Chapter 2.)

[14] I.e.: for all m, w, a: $|^\vee{}^\wedge\alpha|^{m,w,a} \Leftrightarrow |\alpha|^{m,w,a}$. It does *not* hold that $^\wedge{}^\vee\alpha \Leftrightarrow \alpha$.

partialize our semantics, a move motivated by various linguistic considerations including the semantics of so-called attitude reports in natural language.

2 Categories

According to Ajdukiewicz (1935), natural language expressions exhibit a function/argument structure, an aspect he formalized by assigning the 'functor' category (b, a) to expressions which need an expression of category b to yield an expression of category a, and stating the axioms $(b, a), b \vdash a$ and $b, (b, a) \vdash a$, which express that both a sequence consisting of an expression of category (b, a) followed by an expression of category b and a sequence consisting of an expression of category (b, a) preceded by an expression of category b are of category a. So, functors can find their arguments on both sides. Note, however, that in the syntax of natural languages the functors usually have a fixed direction in which they look for their arguments. Hence Bar-Hillel (1953) proposed a *directed* variant, the so-called Ajdukiewicz/Bar-Hillel calculus (**AB**), where right-looking and left-looking functors each have their own axiom, viz., $a/b, b \vdash a$ and $b, b\backslash a \vdash a$, respectively. When we add a so-called *Cut* rule: if $T \vdash a$ and $U, a, V \vdash c$, then $U, T, V \vdash c$ (where a, b denote categories and T, U, V finite sequences of categories, of which T is non-empty), we can use the calculus **AB** for language recognition. Suppose, for example, that we have the following lexical category assignments:

- *Hesperus* $\mapsto$ np,
- *prints* $\mapsto$ $(np\backslash s)/np$,
- $a \mapsto np/n$, and
- *stack* $\mapsto$ n.

Then we can show that *Hesperus prints a stack* is a sentence (s) by proving that $np, (np\backslash s)/np, np/n, n \vdash s$ is a theorem:

(2)
$$\dfrac{np/n, n \vdash np \qquad \dfrac{(np\backslash s)/np, np \vdash np\backslash s \quad np, np\backslash s \vdash s}{np, (np\backslash s)/np, np \vdash s}\,[Cut]}{np, (np\backslash s)/np, np/n, n \vdash s}\,[Cut]$$

More formally:

(3) Let ATOM be a finite set of atomic categories (e.g., $\{np, s, n\}$). Then CAT, the set of *categories* based on ATOM, is the smallest set such that (*i*) ATOM $\subseteq$ CAT, and (*ii*) if $a \in$ CAT and $b \in$ CAT, then $a/b \in$ CAT and $b\backslash a \in$ CAT.

(4) A *sequent* is an expression $T \vdash c$, where T is a finite non-empty sequence of categories and $c \in$ CAT. (So, $T = c_1, \ldots, c_n$, where $n > 0$ and for all i such that $1 \leq i \leq n$: $c_i \in$ CAT.)

We will call $c_1, \ldots, c_n$ the *left-hand side*, and c the *right-hand side* or *goal* of a sequent $c_1, \ldots, c_n \vdash c$.

The calculus **AB** is the smallest set of sequents such that:

(5) (*i*) $\{a/b, b \vdash a \mid a \in \text{CAT} \text{ and } b \in \text{CAT}\} \subseteq \mathbf{AB}$;
(*ii*) $\{b, b\backslash a \vdash a \mid a \in \text{CAT} \text{ and } b \in \text{CAT}\} \subseteq \mathbf{AB}$; and
(*iii*) if $T \vdash a \in \mathbf{AB}$ and $U, a, V \vdash c \in \mathbf{AB}$, then $U, T, V \vdash c \in \mathbf{AB}$.

Let V be a set of lexical items, let Γ be a function which assigns a finite set of categories to each member of V, and let c be a category. Then the *language* recognized by an **AB**-grammar $G = \langle V, \Gamma, c \rangle$ is the set of strings $s_1 \ldots s_n$ over V such that for all i, $1 \leq i \leq n$, there are $c_i \in \Gamma(s_i)$ such that $c_1, \ldots, c_n \vdash c \in \mathbf{AB}$.

Notice that **AB** grammars are extremely lexicalist. We only have universal axioms, a universal *Cut* rule, and a lexicon. There are no fragment-specific rules for combining expressions, since the combinatorial properties of expressions are completely determined by the categories they are assigned in the lexicon. Bar-Hillel, Gaifman and Shamir (1960) proved that **AB** grammars have the weak generative capacity of context-free grammars: both types of grammars recognize exactly the same set of languages.

In 1958, Lambek proposed an crucial extension of **AB**. (The product-free part of) Lambek's calculus, **L**, consists of a set of axioms and five inference rules:

(6) AXIOM, the set of *axioms* of **L**, is the set $\{c \vdash c \mid c \in \text{CAT}\}$.

The *inference rules* of **L** are $/L$, $\backslash L$, $/R$, $\backslash R$ and *Cut*. In (7), a, b, c represent categories, and T, U, V finite sequences of categories, of which T is non-

empty:

$$(7)\quad \frac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c}\,[/L] \qquad \frac{T \vdash b \quad U, a, V \vdash c}{U, T, b\backslash a, V \vdash c}\,[\backslash L]$$

$$\frac{T, b \vdash a}{T \vdash a/b}\,[/R] \qquad \frac{b, T \vdash a}{T \vdash b\backslash a}\,[\backslash R]$$

$$\frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$$

It is easy to see that $\mathbf{L} - \{[/R], [\backslash R]\}$ is equivalent to $\mathbf{AB} \cup$ AXIOM. Observe, on the other hand, that the introduction of $[/R]$ and $[\backslash R]$ essentially enlarges the set of theorems. For example, the calculus **L** allows the derivation of the following theorems, which have been proposed independently by various authors in the sixties and seventies as additional axioms for strengthening the basic (**AB**) framework of categorial grammar:

- $a \vdash b/(a\backslash b)$ (raising)
- $a\backslash(c/b) \vdash (a\backslash c)/b$ (associativity)
- $a/b, b/c \vdash a/c$ (composition)
- $a/b \vdash (a/c)/(b/c)$ (division)

An important result is due to Buszkowski (1986), who proved that the calculus in (7) is complete (and **AB**, therefore, incomplete) with respect to the intuitive interpretation of categories as sets of expressions. That is, let S denote the set of finite sequences of lexical items, let $v \in [c]$ iff $c \in \Gamma(v)$ for lexical items $v \in V$, and let

- $[b\backslash a] = \{s \in \mathrm{S} \mid \text{for all } s' \in [b]\colon\ s'\,s \in [a]\}$,
- $[a/b] = \{s \in \mathrm{S} \mid \text{for all } s' \in [b]\colon\ s\,s' \in [a]\}$, while
- $[c_1] \ldots [c_n] = \{s_1 \ldots s_n \in \mathrm{S} \mid s_1 \in [c_1] \text{ and } \ldots \text{ and } s_n \in [c_n]\}$.

Then the following holds: $[c_1] \ldots [c_n] \subseteq [c]$ if and only if $c_1, \ldots, c_n \vdash c$ is a theorem of **L**. Moreover, Zielonka (1981) has shown that no addition of a finite number of axioms to **AB** can lead to a complete system. This completeness property does not affect the weak generative capacity of the

calculus: Pentus (1992) proved that **L** grammars recognize the same set of languages as **AB** grammars and context-free grammars.

Another important result is already shown in Lambek (1958). He proved that *Cut* is a derived inference rule – the set of theorems is not increased by adding it to the other four inference rules. Since the number of occurrences of / and \ in the premises of the rules $/L$, $\backslash L$, $/R$ and $\backslash R$ is strictly smaller than that in the conclusion, establishing the derivability of the premise sequents is a more simple task than establishing the derivability of the conclusion, and it follows that every sequent has only finitely many *Cut*-free derivations. Note that this entails the decidability of **L**: for any sequent $T \vdash c$, the proof procedure is guaranteed to terminate after finitely many steps with an answer to the question whether the sequent is derivable.

3 Types and Categories

The first gathering of syntactic categories and semantic types took place in the work of Montague. His most influential paper, 'The Proper Treatment of Quantification in Ordinary English' (1973) presents a grammar for a fragment of English which contains a semantic component in which the English expressions defined by the syntactic component are translated into expressions of a language of typed logic. These translations receive their model-theoretic interpretation in the way specified in Section 1 above. Thus the English expressions are assigned a semantic interpretation via the logical expressions which serve as their translations.

All expressions defined by the syntactic component belong to categories. For these categories, Montague uses a notation familiar from categorial grammar: besides basic categories, there are compound categories a/b and $a//b$. The syntactic category of an expression determines the semantic type of its translation (and hence the type of its interpretation): basic categories c are assigned some type $\text{TYPE}(c)$, and compound categories a/b and $a//b$ are assigned the function type $(\text{TYPE}(b), \text{TYPE}(a))$.[15] Note that the category-to-type assignment TYPE is a *function*: every category is associated with a single type. This means that if an expression belongs to category c, then its translation has to be of type $\text{TYPE}(c)$, the unique type assigned to c.

As a consequence, a strategy of 'generalizing to the worst case' has to

[15] In fact, the intensional type $((s, \text{TYPE}(b)), \text{TYPE}(a))$ is assigned to such categories, but the semantics in the present section has been kept extensional for expository purposes. (The true story will be told in Chapter 1 below.)

be adopted. *All* expressions of a certain syntactic category will have to be uniformly assigned a translation of the most complex type that is needed for *some* expression in that category. The proper name *John*, for instance, belongs to the same category as the expression *a personal computer*. But because the latter noun phrase requires an interpretation of type $((e,t),t)$ (cf. Section 1 above), the former one cannot simply be translated as a constant j_e which denotes the individual John. Instead, Montague provides it with the $((e,t),t)$-type translation $\lambda P_{(e,t)}[P_{(e,t)}(j_e)]$, a term which denotes the set of sets that have John as a member.

This aspect of 'orthodox' Montague grammar has been criticized by Partee and Rooth (1983), among others. They argue that the availability of scope ambiguities in natural languages shows that the worst case cannot always be generalized to, since it does not invariably yield the right results.

In the context of the syntax of formal logical languages the notion of scope arises as follows. Such languages typically contain operators O which combine with an expression α to form a compound expression $O\,\alpha$, where α is called the *scope* of O. Variable-binding operators are operators O which combine with a variable v and an expression α (in which v may occur freely) to form the compound expression $Ov\,\alpha$. These operators are said to *bind* the free occurrences of v in their scope α (if any). Well-known examples of variable-binding operators are the existential and universal quantifier in first order logic and the lambda operator in typed logic. Interestingly, scope and binding have also been claimed to be relevant notions for the semantic analysis of natural languages. Within various formal semantic theories certain ambiguities are analyzed in terms of the relative scope of certain operators. For example, the sentence *A Mac adorns every desktop* can be construed in two ways:

- 'There is a Mac that adorns every desktop', and
- 'Every desktop is adorned by a Mac'.

We will see shortly that this so-called 'scope ambiguity' can be characterized in terms of the relative scope of the operators which constitute the interpretations of the noun phrases *a Mac* and *every desktop*. Moreover, binding of variables by such operators can be invoked in representing the relation between anaphoric pronouns and their antecedents. A sentence like *Every Mac says that it has unexpectedly quit* can be analyzed in terms of a variable-binding operator 'For every Mac x:' which binds two free occur-

rences of the variable x in the expression 'x says that x has unexpectedly quit':

- 'For every Mac x: x says that x has unexpectedly quit'

The grammar of Montague (1973) accounts for the above phenomena in a way which is consistent with 'Frege's principle' – also known as the compositionality principle –, according to which the meaning of an expression is a function of the meanings of its parts and of the way in which these parts are syntactically combined.

The syntactically minimal parts of the sentence *Everyone admires someone*, for example, are *everyone*, *someone*, and *admires*. Somewhat simplified, the noun phrases *everyone* and *someone* are translated as the terms $\lambda P_{(e,t)} \forall x_e[P_{(e,t)}(x_e)]$ and $\lambda P_{(e,t)} \exists y_e[P_{(e,t)}(y_e)]$ of type $((e,t),t)$, which denote the set of sets that contain all individuals and the set of sets that contain some individual, respectively. The verb *admires* is interpreted as the term $\lambda T_{((e,t),t)} \lambda v_e . T_{((e,t),t)}(\lambda w_e . \text{ADMIRE}_{(e,(e,t))}(w_e)(v_e))$ of type $(((e,t),t),(e,t))$, a ('worst case') translation which denotes a relation which holds between a set of sets of individuals and an individual if and only if the set of individuals admired by the individual is a member of the set of sets of individuals.

A rule which constructs verb phrases says that their translation is obtained by applying the translation of the transitive verb to the translation of the direct object. For the verb phrase [$_{\text{VP}}$ *admires someone*], this results in the term $[\lambda T \lambda v . T(\lambda w . \text{ADMIRE}(w)(v))](\lambda P \exists y[P(y)])$ of type (e,t), which reduces to $\lambda v \exists y \, \text{ADMIRE}(y)(v)$ after three λ-conversions.

A rule building up sentences interprets them as the result of applying the translation of the subject to the translation of the verb phrase. The outcome for the sentence [$_{\text{S}}$ *everyone*[$_{\text{VP}}$ *admires someone*]] is the term $[\lambda P \forall x[P(x)]](\lambda v . \exists y \, \text{ADMIRE}(y)(v))$ of type t. After two λ-conversions, this term reduces to $\forall x \exists y \, \text{ADMIRE}(y)(x)$.

This way of constructing sentences always yields readings in which the subject takes scope over the direct object. However, the sentence *Everyone admires someone* is assumed to have a stronger reading as well, $\exists y \forall x \, \text{ADMIRE}(y)(x)$, in which the direct object noun phrase has wide scope. Notice that this 'scope ambiguity' is not obviously of a lexical or structural nature, since the lexical meanings and the syntactic structure involved do not seem to be different. Hence, given Frege's principle, a second way of constructing the same syntactic structure is required.

For that purpose, Montague introduces an infinite number of so-called syntactic variables $he_0, he_1, he_2, \ldots$, noun phrases which are translated as

$\lambda P_{(e,t)}[P_{(e,t)}(v_{0,e})], \lambda P_{(e,t)}[P_{(e,t)}(v_{1,e})], \lambda P_{(e,t)}[P_{(e,t)}(v_{2,e})], \ldots$, respectively, as well as an infinite number of rules of quantifying-in: $Q_0, Q_1, Q_2, \ldots$ Syntactically, the Q_i are rules of substitution: if α is a noun phrase and β is a sentence, then $Q_i(\alpha, \beta)$ is a sentence, and $Q_i(\alpha, \beta)$ results from β by replacing the first occurrence of he_i by α, and the remaining occurrences by appropriate anaphoric pronouns. The translation of the sentence $Q_i(\alpha, \beta)$ is obtained by applying α', the translation of α, to $\lambda x_i \beta'$, where β' is the translation of β. That is: $Q_i(\alpha, \beta)$ translates as $\alpha'(\lambda x_i \beta')$.

With these devices we can construct *Everyone admires* he_{17}, which is interpreted as $\forall x\,\text{ADMIRE}(x_{17})(x)$. Applying Q_{17} to that 'sentence' and the noun phrase *someone* give us again *Everyone admires someone*, this time interpreted as $[\lambda P \exists y[P(y)]](\lambda x_{17}\,\forall x\,\text{ADMIRE}(x_{17})(x))$, which reduces to $\exists y \forall x\,\text{ADMIRE}(y)(x)$ after two λ-conversions.

It can be argued that in this particular case the 'object wide scope' interpretation needs no separate semantic representation, since it is stronger than the 'wide scope subject' interpretation: if the latter is true, then the former is true as well. However, there are sentences like *A Mac adorns every desktop*, where the situation is the other way round. Moreover, quantifying-in can also take care of logically unrelated readings. E.g., if the verb *seeks* is translated as SEEK of type $(((e,t),t),(e,t))$, then the sentence *Find File seeks a document* gets the *de* -reading $\text{SEEK}(\lambda P \exists x[\text{DOCUMENT}(x) \wedge P(x)])(\textit{ff})$ without quantifying-in. Using quantifying-in, however, one obtains the logically independent *de re*-reading $\exists x[\text{DOCUMENT}(x) \wedge \text{SEEK}(\lambda P[P(x)])(\textit{ff})]$.

In addition to the representation of quantifier scope, rules of quantifying-in are used for an analysis of anaphoric pronouns in terms of binding. Thus, he_4 *says that* he_4 *has unexpectedly quit* translates as $\text{SAY}(\text{H.U.QUIT}(x_4))(x_4)$. If we use Q_4 to combine this sentence with the noun phrase *every Mac* which translates as $\lambda P \forall x[\text{MAC}(x) \rightarrow P(x)]$, we arrive at the sentence *Every Mac says that it has unexpectedly quit*, and its translation reduces to the term $\forall x[\text{MAC}(x) \rightarrow \text{SAY}(x, \text{H.U.QUIT}(x))]$.

There exist, of course, alternatives to Montague's treatment of quantification and bound anaphora. Within the framework of model-theoretic semantics, Cooper (1975, 1983) has proposed a mechanism for semantically storing quantifiers and pronouns. His approach avoids the unintuitive syntactic features of quantifying-in, but this is accomplished at the expense of complicating the semantic component.

Chapter 1 of the present dissertation proposes flexible interpretation as an alternative which avoids the unintuitive syntactic and semantic features of quantifying-in and storage. This alternative involves giving up Montague's

strategy of uniformly assigning *all* members of a certain category the most complicated type that is needed for *some* expression in that category. It will be argued that the strategy of generalizing to the worst case fails, not because the worst case cannot always be generalized to, but simply because there *is* no such case. Instead, a reverse strategy is proposed which generalizes to the 'best case' on the lexical level, since every lexical expression is assigned a basic translation of the minimal type available for that expression. Generalized syntactic/semantic rules permit the compounding of all 'mutually fitting' translations, type-shifting rules produce derived translations out of lexical and compound ones, and the recursive nature of these rules reflects the empirical fact that there is no worst case.

It should be noted here that flexible interpretation shares some shortcomings with quantifying-in and storage. All proposals analyse quantifier scope ambiguities in terms of different linearly ordered scope relations between quantified noun phrase interpretations. Thus, the combination of an $(e,(e,t))$-type transitive verb translation TV with the $((e,t),t)$-type translations NP_2 of a quantified object noun phrase and NP_1 of a quantified subject noun phrase will give rise to the following two readings:

(*a*) $\mathrm{NP}_1(\lambda y_e.\mathrm{NP}_2(\lambda x_e.\mathrm{TV}(x)(y)))$ (*subject wide scope*)

(*b*) $\mathrm{NP}_2(\lambda x_e.\mathrm{NP}_1(\lambda y_e.\mathrm{TV}(x)(y)))$ (*object wide scope*)

This saddles these approaches with two kinds of problems.

First, they predict too much, since there are cases where object wide scope readings are obviously absent. For instance, a sentence like *Zero laserwriters print zero documents* does have the subject wide scope reading 'Every laserwriter prints at least one document', but it cannot mean that every document is printed by at least one laserwriter.

And second, they predict too little. As Scha (1981) has pointed out, a sentence such as *600 Dutch firms own 5000 American computers* has a so-called cumulative reading, in which the noun phrases have 'equally wide' scope: 'The number of Dutch firms owning an American computer is 600, and the number of American computers owned by a Dutch firm is 5000'. (Note that an analogous interpretation is available for the sentence *Zero laserwriters print zero documents*.) In an important paper (1992), Keenan has proven that cumulative readings, as well as a number of readings of sentences involving particular quantifiers, such as the context-independent

reading[16] of *Every firm owns a different computer*, are 'beyond the Frege boundary': they are just not expressible in terms of (*a*) or (*b*).

In Chapter 2 of the present dissertation it is shown that flexible interpretation also does not cross the Frege boundary in that it observes the compositionality priniple. A detailed consideration of the implications of the principle of compositionality for the organization of grammar fragments in general leads to a formalization of the principle which differs from the one presented in Janssen (1986). It is argued that this formalization can be motivated and applied more easily, and that it avoids some technical complications inherent in Janssen's approach. The flexible Montague grammar proposed in Chapter 1 turns out to be compositional under the 'most intuitive' interpretation of the principle, provided that the type-shifting derivation of translations is explicitly incorporated into the grammar.

We noted above that various proposals were made in the sixties and seventies for strengthening the basic (**AB**) framework of categorial grammar. As a matter of historical fact, most of these proposals were already present in the calculus **L** that was introduced in Lambek (1958). After 1980, the rediscovery of this paper led to a renaissance of categorial research which was also inspired by Montague's work on the semantics of natural language.[17] In view of the central role played by function/argument structure, categorial grammar is a formalism which can accommodate both syntactic categories and semantic types. And in fact, Van Benthem (1986, Chapter 7; 1991) showed that both perspectives can be systematically related in the Lambek calculus: the semantic interpretations of an **L**-derivable sequent are directly determined by the proofs of its validity in the syntactic calculus.

In Moortgat's (1988) adaptation of Van Benthem's term construction algorithm for the natural deduction formulation of **L** to the sequent calculus, every category occurrence c in a derivation is assigned some typed lambda term γ. The type of γ depends on c. For categories $c \in \text{ATOM}$, some type assignment $\text{TYPE}(c)$ is assumed, and the function TYPE is extended to non-atomic categories by the stipulation that $\text{TYPE}(a/b) = \text{TYPE}(b\backslash a) = (\text{TYPE}(b), \text{TYPE}(a))$. In this semantics, which is given in (8) below, the

[16] 'The own-relation restricted to firms on its first argument and to computers on its second, is one-to-one', to be contrasted with the context-dependent reading which is suggested by *John owns a Mac, but every firm owns a different computer* (cf. Keenan (1987) and Van Benthem (1989)).

[17] Cf. Oehrle, Bach and Wheeler (1988) and Buszkowski, Marciszewski and Van Benthem (1988). The logical foundations of this work are studied in Van Benthem (1991).

axioms of **L** correspond to *identity*, the rules $/L$ and $\backslash L$ to *application*, the rules $/R$ and $\backslash R$ to *abstraction*, and the *Cut* rule to *substitution*.[18] The respective assignment of terms $\gamma_1, \ldots, \gamma_n, \gamma$ to categories $c_1, \ldots, c_n, c$ in a derivable sequent $c_1, \ldots, c_n \vdash c$ is written as $c_1 : \gamma_1, \ldots, c_n : \gamma_n \vdash c : \gamma$. In (8), only the terms of the 'active' categories are indicated separately. The expression v denotes a variable of type TYPE(b), while α, β, γ and φ denote terms of type TYPE(a), TYPE(b), TYPE(c), and (TYPE(b), TYPE(a)), respectively. The expressions T', U' and V' refer to the sequences of terms assigned to the sequences of 'inactive' categories T, U, and V. Thus, the terms assigned to the categories in T, U, V are identical in premise and conclusion.

(8)
$$c : \gamma \vdash c : \gamma \quad [\text{AXIOM}]$$

$$\frac{T : T' \vdash b : \beta \qquad U : U', a : \varphi(\beta), V : V' \vdash c : \gamma}{U : U', a/b : \varphi, T : T', V : V' \vdash c : \gamma}\,[/L]$$

$$\frac{T : T' \vdash b : \beta \qquad U : U', a : \varphi(\beta), V : V' \vdash c : \gamma}{U : U', T : T', b\backslash a : \varphi, V : V' \vdash c : \gamma}\,[\backslash L]$$

$$\frac{T : T', b : v \vdash a : \alpha}{T : T' \vdash a/b : \lambda v.\alpha}\,[/R] \qquad \frac{b : v, T : T' \vdash a : \alpha}{T : T' \vdash b\backslash a : \lambda v.\alpha}\,[\backslash R]$$

$$\frac{T : T' \vdash a : \alpha \qquad U : U', a : \alpha, V : V' \vdash c : \gamma}{U : U', T : T', V : V' \vdash c : \gamma}\,[Cut]$$

Chapter 3 of the present dissertation presents and motivates an alternative, equivalent formulation of the Van Benthem/Moortgat semantics for **L**, and uses it in a straightforward semantic version of Lambek's *Cut* elimination theorem which entails that **L** is semantically decidable as well: the result of applying Lambek's *Cut* elimination algorithm is a derivation which is semantically equivalent to the original derivation. Moreover, it is shown that **L** can be further normalized to a calculus **L*** that offers a solution to the so-called 'spurious ambiguity problem' – the problem that different proofs of a given sequent may yield one and the same semantic interpretation: each interpretation of a sequent corresponds to exactly one proof in **L***. This solution is compared with the proposals in Moortgat (1990) and Roorda (1991) and applied in an extension of an encoding result of Ponse (1988).

We note, finally, that the calculus **L** is in itself an elegant but elementary system: it is only a weak, non-commutative implicational fragment of

[18] Viz., the substitution of T' (which yields α) for α in $U', \alpha, V' \vdash \gamma$.

linear logic (cf. Girard (1987)). Accordingly, recent research in categorial grammar has been heading towards the extension of this basic system with new type-forming operators, to be used in analyses of discontinuity, locality, left-incremental parsing, non-constituent coordination and other phenomena (see Moortgat (1988), Barry and Morrill (eds.) (1990), Hepple (1990), Emms (1992), Morrill (1990a; 1990b; 1992; 1993), Moortgat and Morrill (1992)).

Chapter 1

Flexible Montague Grammar

> For one thing, our principles can be applied more flexibly
> so as to explain certain multiple ambiguities ...
>
> Jaakko Hintikka, *Quantifiers vs. Quantification Theory*

In this chapter I will argue that adoption of flexible type assignment in Montague grammar leads to an elegant account of natural language scope ambiguities which arise in the presence of quantifying and coordinating expressions.

Whereas Montague's original fragments (1970a; 1970b; 1973) resort to the syntactic device of quantifying-in for representing quantifier scope ambiguities, Cooper's alternative mechanism (1975; 1983) of semantically storing quantifiers avoids the 'unintuitive' syntactic aspects of Montague's proposal – at the expense, however, of complicating the semantic component. Hence Cooper's conclusion:

> the syntactic treatment of wide scope quantification [...] does not seem satisfactory since it involves a rule [...] which is not syntactically motivated [...]. The alternative takes the complication out of the syntax and puts it into the semantics by introducing storage interpretation rules. The fact of the matter is, that however we do wide scope quantification it seems to involve somewhat unpalatable principles either in the syntax or in the semantics. (Cooper (1983), p. 65)

Below it will be shown that flexible type assignment leads to a more adequate division of labour between the syntactic and semantic component, in that it renders both quantifying-in and storage superfluous. Any remaining unpalatability will have to be located in the relationship between the two components.

The chapter is organized as follows. In Section 1, 'Syntax and Semantics', the relevant properties of orthodox Montague grammar are reviewed, and Montague's and Cooper's approaches to scope ambiguities are outlined.

Next, Section 2, 'Arguments for Flexibility', lists arguments that can be (and have been) put forward in favour of a less rigid category-to-type assignment. The consequences of the apparent need of flexibility for the design of the grammar are first explored in Section 3, 'Flexibility in Syntax and Semantics', in which we consider the Lambek calculus, an influential theory of syntactic and semantic flexibility. Section 4, 'The Fragment', then presents a fragment of flexible Montague grammar, which is best seen as creating a 'logical space' of scope possibilities which is not necessarily exhausted by the facts of particular natural languages. Section 5, 'Applications', shows that this flexible grammar allows one to represent scope ambiguities without special syntactic or semantic devices, and discusses the possibility of implementing structural restrictions on the availability of readings. Finally, Section 6, 'Flexibility and Anaphora', deals with the second aspect of quantifying-in, the representation of anaphoric pronouns, and sketches how it can be accounted for in a flexible set-up. The semantic properties of the proposed fragment are treated in an Appendix.

1 Syntax and Semantics

In his paper 'The Proper Treatment of Quantification in Ordinary English' (Montague (1973), PTQ henceforth), Richard Montague presented a grammar for a fragment of English which contains a semantic component in which the expressions defined by the syntactic component are translated into expressions of **IL** (= Intensional Logic), a logical language of intensional type theory. These expressions of **IL** 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 which are their translations. The 'PTQ fragment' has since acquired a paradigmatic status within the framework of compositional model-theoretic semantics.[1]

In the present section we will go into a number of aspects of the fragment, which will be considered with its original semantics[2] (that is: including

[1]See Janssen (1986), Part 1, for a clear exposition of this framework; less clear, more concise: Montague (1970b). See also the textbooks Dowty, Wall and Peters (1981), and Gamut (1991), Vol. II.

[2]But there are also several – inessential – differences: S replaces Montague's category t, PV abbreviates his IV/t, Montague's categories t/t, IV/IV, IV//IV and (IV/IV)/T have been omitted, determiners are treated non-syncategorematically, there are no 'rules of tense and sign', the set of lexical expressions is different, and the syntax introduces

individual concepts – cf. Dowty, Wall and Peters (1981), Appendix III, and Gamut (1991), Vol. II, Section 6.4.1).

1.1 Categories

The set of syntactic categories in the PTQ fragment is defined inductively.

There are two basic categories, S and E. The category 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 categories as well. (Outer brackets of categories will be omitted in the sequel.) Both A/B and A//B represent categories consisting of expressions which 'take'[3] 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, however, we will only have to deal with a finite subset of the defined categories: the eight categories specified below.

(1)

CATEGORY	DESCRIPTION	DEFINITION
S	sentence	—
E	—	—
CN	common noun	S/E
IV	intransitive verb phrase	S//E
T (NP)	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

labelled brackets. (Relative clauses will be discussed in Section 5.)

[3]Montague employs a notation familiar from categorial grammar. In categorial grammar as well as in PTQ, the syntactic category of an expression is used to determine the semantic type of its interpretation (cf. Section 3 below).

PTQ differs from standard categorial grammar, however, in that it contains fragment-specific rules which give a recursive definition of the set of English expressions. (This is why, for instance, S/E and S//E are distinguished: the denotations of their expressions have the same semantic type, but both categories figure in different rules.) Moreover, these rules take care of much more than mere concatenation of expressions. They involve morphological retrieval, insertion of syncategorematic expressions, substitution, etc.

Hence one cannot simply say that, e.g., expressions of category A/B are the ones which form an expression of category A when they are *concatenated* with an expression of category B – something which, for example, does hold for the Lambek calculus, cf. Section 2 of the Introduction above and footnote 41 below.

With the exception of S and E, all categories contain lexical elements. (The term phrases $[_{\mathrm{T}} he_0]$, $[_{\mathrm{T}} he_1]$, $[_{\mathrm{T}} he_2]$, ... are called *syntactic variables*.)

(2)
CN:	$[_{\mathrm{CN}}$ *man*$]$, $[_{\mathrm{CN}}$ *woman*$]$, $[_{\mathrm{CN}}$ *unicorn*$]$, $[_{\mathrm{CN}}$ *temperature*$]$
IV:	$[_{\mathrm{IV}}$ *walks*$]$, $[_{\mathrm{IV}}$ *rises*$]$
T:	$[_{\mathrm{T}}$ *John*$]$, $[_{\mathrm{T}}$ *Mary*$]$, ..., $[_{\mathrm{T}} he_0]$, $[_{\mathrm{T}} he_1]$, $[_{\mathrm{T}} he_2]$, ...
Det:	$[_{\mathrm{Det}}$ *the*$]$, $[_{\mathrm{Det}}$ *every*$]$, $[_{\mathrm{Det}}$ *a*$]$, $[_{\mathrm{Det}}$ *one*$]$
TV:	$[_{\mathrm{TV}}$ *finds*$]$, $[_{\mathrm{TV}}$ *seeks*$]$
PV:	$[_{\mathrm{PV}}$ *claims that*$]$, $[_{\mathrm{PV}}$ *believes that*$]$

1.2 Types

There is a *rigid* correspondence between syntactic categories and semantic types in PTQ. Thus, if an expression α belongs to a certain category C, then its **IL** translation α' is invariably of the unique type assigned to C: TYPE(C). Put differently, TYPE, the category-to-type assignment in PTQ, is a *function*. This function is defined as follows: TYPE(S) = t, TYPE(E) = e, and TYPE(A/B) = TYPE(A//B) = $((s, \mathrm{TYPE(B)}), \mathrm{TYPE(A)})$. 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*, type e the set of *entities*, and type s (which is actually not an independent, basic type in **IL**) the set of *possible worlds*.

A compound type (a, b) stands for the set of *functions* from objects of type a to objects of type b. Whenever b equals t, we can identify such a function of type $(a, b) = (a, t)$ with a *set of objects* of type a. And when the type is of form $(a, (b, t))$, we can identify an object in its denotation with a *set of ordered pairs* $\langle \alpha, \beta \rangle$, where α is an object of type a, and β is an object

of type b; hence such objects correspond to binary *relations* between objects of type a and objects of type b.

Correspondingly, by stipulating that TYPE(S) equals t, we establish that the denotation of a sentence is a truth value. If there were expressions of category 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* being a function from possible worlds to entities. (An object of type $((s,e),t)$ is a function from objects of type (s,e) to objects of type t. Such a function can be identified with a set of objects of type (s,e), i.e., a set of functions from possible worlds to entities: 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 a being an object of type $(s,(a,t))$ – a function from possible worlds to sets of objects of type a. 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* being a function from possible worlds to truth values.

The rigid correspondence between categories and types has some consequences. Notice, 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 T, the category of noun phrases, which *John* belongs to, is associated with type $((s,((s,e),t)),t)$ by the function TYPE, and not with e. Therefore, we are obliged to translate *John* into an expression of the former type. Montague chose $\lambda P.^{\vee}P(^{\wedge}j)$, an expression which refers to the set of those properties of individual concepts the extension of which contains the intension of the denotation of j as their element, and this solution turns out to work.

Now, 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 quantified noun phrases like *every woman* (which cannot be associated with type e for obvious semantic reasons) have a very similar distribution, they can be coordinated (*John and every woman*), etc. Such an assignment would, moreover, complicate the grammar to a great extent: every rule which applies both to proper names and to quantified noun phrases would have to be doubled.

1.3 Translation Rules

The rules of the PTQ grammar consist of a syntactic part and a semantic part. The syntax gives a recursive definition of the set of expressions which constitute a fragment of English. In our set-up, the basic expressions are defined by the lexicon in (2), which contains labelled bracketings like (4):

(4) $[_{\mathrm{CN}}\mathit{man}]$

The meaning of (4) is that *man* is a common noun. Furthermore, there are syntactic rules. For example:

(5) $[_{\mathrm{T}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow [_{\mathrm{S}}[_{\mathrm{T}}\alpha][_{\mathrm{IV}}\beta]]$

This rule says, roughly speaking,[4] that α β is a sentence if α is a noun phrase and β an intransitive verb phrase. We will ignore further details of syntactic analysis.

Semantic translation rules specify (*i*) the interpretation (strictly speaking: the **IL** translation) of the lexical expressions; and (*ii*) the interpretation of compound expressions, *given the interpretations of their 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 expression of **IL** 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 (1991), Section 6.4.1 of Volume II, for arguments), but not for 'normal' common nouns like *man*, *woman*, *unicorn*, which intuitively denote sets of individuals. That is why Montague supplemented his translations of these common nouns with a meaning postulate (MP 2 in PTQ). In doing so, he accounted for the fact that such 'normal' common nouns denote sets of *constant* individual concepts (that is: objects which are equivalent in some sense to sets of individuals).

We opt here for a slightly different approach, and incorporate Montague's meaning postulates into the lexical translations (ignoring Montague's

[4]Strictly speaking, one has to distinguish between labelled bracketings and their expressions. (The latter are obtained from the former by deleting all occurrences of $[_{\mathrm{S}}$, $[_{\mathrm{CN}}$, $[_{\mathrm{IV}}$, $[_{\mathrm{T}}$, $[_{\mathrm{Det}}$, $[_{\mathrm{TV}}$, $[_{\mathrm{PV}}$, and $]$.) A syntactic rule $[_{\mathrm{A}}\alpha] + [_{\mathrm{B}}\beta] \rightarrow [_{\mathrm{C}}\gamma]$ expresses that γ's expression is a C whenever α's expression is an A and β's expression is a B.

*-notation) – cf. (6) below. The effects of meaning postulate 2 are reflected in our translations of *man*, *woman* and *unicorn*.

Analogously, Montague's meaning postulate 3, for IVs, is hidden in the translation of *walks*. This intransitive verb holds of the *value* of the individual concepts that constitute its denotation: the subject position of such verbs is extensional. The meaning postulate does not hold for *rises*. Consequently, this verb is simply translated as the non-logical constant RISE of type $((s,e),t)$.

We have incorporated PTQ's meaning postulates 5 and 6 into the translation of *seeks* and *finds*, and *claims that* and *believes that*, respectively. They account for the extensionality of the subject position of TVs and PVs.

In the case of *finds*, the object position is extensional as well, which explains why the translation of this verb also embodies Montague's meaning postulate 4.[5]

(6)

$[_{\text{CN}}$*temperature*$]$	TEMPERATURE
$[_{\text{CN}}$*man*$]$	$\lambda x.[\text{MAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$
$[_{\text{CN}}$*woman*$]$	$\lambda x.[\text{WOMAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$
$[_{\text{CN}}$*unicorn*$]$	$\lambda x.[\text{UNICORN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$
$[_{\text{IV}}$*rises*$]$	RISE
$[_{\text{IV}}$*walks*$]$	$\lambda x.\text{WALK}(^{\vee}x)$
$[_{\text{T}}$*John*$]$	$\lambda P.{}^{\vee}P(^{\wedge}j)$
$[_{\text{T}}$*Mary*$]$	$\lambda P.{}^{\vee}P(^{\wedge}m)$

[5]The constants in (6) have the following types: $\text{TEMPERATURE}_{((s,e),t)}$, $\text{MAN}_{(e,t)}$, $\text{WOMAN}_{(e,t)}$, $\text{UNICORN}_{(e,t)}$, $\text{RISE}_{((s,e),t)}$, $\text{WALK}_{(e,t)}$, j_e, m_e, $\text{SEEK}_{((s,((s,((s,e),t)),t)),(e,t))}$, $\text{FIND}_{(e,(e,t))}$, $\text{CLAIM}_{((s,t),(e,t))}$, $\text{BELIEVE}_{((s,t),(e,t))}$. The variables in (6) are typed as follows: $x_{(s,e)}$, $y_{(s,e)}$, $x_{i\,(s,e)}$, u_e, v_e, $P_{(s,((s,e),t))}$, $Q_{(s,((s,e),t))}$, $T_{(s,((s,((s,e),t)),t))}$, $p_{(s,t)}$.

In Montague's definition of the language **IL**, the 'real' variables constitute the family of sets $\{\{v_{n,\tau} \mid n \in \mathbb{N}\} \mid \tau \text{ is a type}\}$, whereas the 'variables' x, y, x_i, u, v, P, Q, T, and p in (6) are abbreviations of $v_{1,(s,e)}$, $v_{3,(s,e)}$, $v_{2i,(s,e)}$, $v_{0,e}$, $v_{1,e}$, $v_{0,(s,((s,e),t))}$, $v_{1,(s,((s,e),t))}$, $v_{0,(s,((s,((s,e),t)),t))}$, and $v_{0,(s,t)}$, respectively.

Since they 'really' carry even subscripts (viz., $2i$), the variables x_i of type (s,e) which occur freely in translations of (expressions containing) syntactic variables are all different from the variables x and y of type (s,e) which occur bound in the translation rules (I) through (XII),i of Section 1.3.2: x and y have odd subscripts. Therefore, the x_i are never 'accidentally' bound by operators containing x and y. (For example, if λx in rule (VII) could bind such a free variable x_i, then undesirable interpretations such as:

$$[\lambda P.{}^{\vee}P(^{\wedge}m)](^{\wedge}\lambda x.[\lambda P.{}^{\vee}P(^{\wedge}j)](^{\wedge}\lambda x.[\text{WALK}(x) \wedge \text{FIND}(x)(x)])),$$

'John walks and finds himself', for *John walks and finds Mary* would result, viz., from applying rule (X),i to *Mary* and *John walks and finds he*$_i$ (with $i = \frac{1}{2}$, so to speak).) The variables x_i can only get bound by the λx_i-operator in (X),i, (XI),i and (XII),i, but this is (of course) intended. See also footnote 20 below.

$[_{\mathrm{T}} he_i]$	$\lambda P.{}^{\vee}P(x_i)$
$[_{\mathrm{Det}}\, a]$	$\lambda P\lambda Q.\exists x[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)]$
$[_{\mathrm{Det}}\, every]$	$\lambda P\lambda Q.\forall x[{}^{\vee}P(x) \rightarrow {}^{\vee}Q(x)]$
$[_{\mathrm{Det}}\, the]$	$\lambda P\lambda Q.\exists x[\forall y[{}^{\vee}P(y) \leftrightarrow x = y] \wedge {}^{\vee}Q(x)]$
$[_{\mathrm{Det}}\, one]$	$\lambda P\lambda Q.\exists x\forall y[[{}^{\vee}P(y) \wedge {}^{\vee}Q(y)] \leftrightarrow x = y]$
$[_{\mathrm{TV}}\, seeks]$	$\lambda T\lambda x.\mathrm{SEEK}(T)({}^{\vee}x)$
$[_{\mathrm{TV}}\, finds]$	$\lambda T\lambda x.{}^{\vee}T({}^{\wedge}\lambda y.\mathrm{FIND}({}^{\vee}y)({}^{\vee}x))$
$[_{\mathrm{PV}}\, claims\ that]$	$\lambda p\lambda x.\mathrm{CLAIM}(p)({}^{\vee}x)$
$[_{\mathrm{PV}}\, believes\ that]$	$\lambda p\lambda x.\mathrm{BELIEVE}(p)({}^{\vee}x)$

1.3.2 Compound Expressions

PTQ contains a number of rules for the translation of compound expressions which belong to the categories mentioned in Section 1.1 above. In what follows α' and β' denote the translations of α and β, respectively.

RULES OF APPLICATION		TRANSLATION
(I)	$[_{\mathrm{T}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow [_{\mathrm{S}}[_{\mathrm{T}}\alpha][_{\mathrm{IV}}\beta]]$	$\alpha'({}^{\wedge}\beta')$
(II)	$[_{\mathrm{TV}}\alpha] + [_{\mathrm{T}}\beta] \rightarrow [_{\mathrm{IV}}[_{\mathrm{TV}}\alpha][_{\mathrm{T}}\beta]]$	$\alpha'({}^{\wedge}\beta')$
(III)	$[_{\mathrm{Det}}\alpha] + [_{\mathrm{CN}}\beta] \rightarrow [_{\mathrm{T}}[_{\mathrm{Det}}\alpha][_{\mathrm{CN}}\beta]]$	$\alpha'({}^{\wedge}\beta')$
(IV)	$[_{\mathrm{PV}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow [_{\mathrm{IV}}[_{\mathrm{PV}}\alpha][_{\mathrm{S}}\beta]]$	$\alpha'({}^{\wedge}\beta')$

RULES OF CONJUNCTION AND DISJUNCTION		TRANSLATION
(V)	$[_{\mathrm{S}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow [_{\mathrm{S}}[_{\mathrm{S}}\alpha]\ and\ [_{\mathrm{S}}\beta]]$	$[\alpha' \wedge \beta']$
(VI)	$[_{\mathrm{S}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow [_{\mathrm{S}}[_{\mathrm{S}}\alpha]\ or\ [_{\mathrm{S}}\beta]]$	$[\alpha' \vee \beta']$
(VII)	$[_{\mathrm{IV}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow [_{\mathrm{IV}}[_{\mathrm{IV}}\alpha]\ and\ [_{\mathrm{IV}}\beta]]$	$\lambda x.[\alpha'(x) \wedge \beta'(x)]$
(VIII)	$[_{\mathrm{IV}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow [_{\mathrm{IV}}[_{\mathrm{IV}}\alpha]\ or\ [_{\mathrm{IV}}\beta]]$	$\lambda x.[\alpha'(x) \vee \beta'(x)]$
(IX)	$[_{\mathrm{T}}\alpha] + [_{\mathrm{T}}\beta] \rightarrow [_{\mathrm{T}}[_{\mathrm{T}}\alpha]\ or\ [_{\mathrm{T}}\beta]]$	$\lambda P.[\alpha'(P) \vee \beta'(P)]$

RULES OF QUANTIFICATION (for $i \in \mathbb{N}$)		TRANSLATION
(X), i	$[_{\mathrm{T}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow [_{\mathrm{S}}\beta^*]$	$\alpha'({}^{\wedge}\lambda x_i.\beta')$
(XI), i	$[_{\mathrm{T}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow [_{\mathrm{IV}}\beta^*]$	$\lambda y.\alpha'({}^{\wedge}\lambda x_i.\beta'(y))$
(XII), i	$[_{\mathrm{T}}\alpha] + [_{\mathrm{CN}}\beta] \rightarrow [_{\mathrm{CN}}\beta^*]$	$\lambda y.\alpha'({}^{\wedge}\lambda x_i.\beta'(y))$

In the rule schemes (X),i, (XI),i and (XII),i, the expression β^* stands for β with (*i*) if $[_{\mathrm{T}}\alpha]$ is a syntactic variable, all occurrences of he_i replaced by α; and (*ii*) if $[_{\mathrm{T}}\alpha]$ is not a syntactic variable, the first occurrence of he_i replaced by α, and the remaining occurrences of he_i replaced by anaphoric pronouns of appropriate case and gender.

1.4 Scope Ambiguities: Quantifying-in

The rule schemes (X),*i*, (XI),*i* and (XII),*i* are used to represent *quantifier scope ambiguities*: ambiguities which arise because there is more than one quantified noun phrase present in a sentence, as well as *de dicto/de re ambiguities*: ambiguities which arise because of the co-occurrence of quantified noun phrases and expressions which give rise to intensional contexts.[6]

For instance, while constructing the sentence *Every man finds a woman* it is possible to use only rules of application. In that case we first apply rule (III) to the lexical translations of *a* and *woman*, cf. (7):

(7) $[\lambda P\lambda Q.\exists x[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)]]({}^{\wedge}\lambda x.[\text{WOMAN}({}^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)])$

We can reduce (7) to (8) by means of λ-conversion and ${}^{\vee}{}^{\wedge}$-elimination:

(8) $\lambda Q.\exists x[[\text{WOMAN}({}^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)] \wedge {}^{\vee}Q(x)]$

The above expression of **IL** can be reduced further to (9), in which process the variable x of type (s, e) is replaced by a variable v of type e.

(9) $\lambda Q.\exists v[\text{WOMAN}(v) \wedge {}^{\vee}Q({}^{\wedge}v)]$

Application of rule (II) to the lexical translation of *finds* and to (9) yields *finds a woman* with translation (10), which reduces to (11).

(10) $[\lambda T\lambda x.{}^{\vee}T({}^{\wedge}\lambda y.\text{FIND}({}^{\vee}y)({}^{\vee}x))]({}^{\wedge}\lambda Q.\exists v[\text{WOMAN}(v) \wedge {}^{\vee}Q({}^{\wedge}v)])$

(11) $\lambda x.\exists v[\text{WOMAN}(v) \wedge \text{FIND}(v)({}^{\vee}x)]$

Rule (III) is also used to build up *every man*, with reduced translation (12).

(12) $\lambda Q.\forall u[\text{MAN}(u) \rightarrow {}^{\vee}Q({}^{\wedge}u)]$

Finally, rule (I) concatenates *every man* and *finds a woman*. Semantically, (12) is applied to the intension of (11). The result, (13), is equivalent to (14): the wide scope-*every* translation of *Every man finds a woman*.

(13) $[\lambda Q.\forall u[\text{MAN}(u) \rightarrow {}^{\vee}Q({}^{\wedge}u)]]({}^{\wedge}\lambda x.\exists v[\text{WOMAN}(v) \wedge \text{FIND}(v)({}^{\vee}x)])$

[6] Another use to which rules of quantification – and quantifier stores – are put is the treatment of *bound pronouns*. This aspect will be discussed in Section 6.

(14) $\forall u[\text{MAN}(u) \rightarrow \exists v[\text{WOMAN}(v) \wedge \text{FIND}(v)(u)]]$

Alternatively, we can first construct the 'sentence' *Every man finds* he_8, the translation of which reduces to (15), and then apply (X),8 (an instantiation of the rule scheme (X),*i*) to *a woman* and *Every man finds* he_8. The semantic result is (16), which reduces to the wide scope-*a* translation (17).

(15) $\forall u[\text{MAN}(u) \rightarrow \text{FIND}(^{\vee}x_8)(u)]$

(16) $[\lambda Q.\exists v[\text{WOMAN}(v) \wedge {}^{\vee}Q(^{\wedge}v)]](^{\wedge}\lambda x_8.\forall u[\text{MAN}(u) \rightarrow \text{FIND}(^{\vee}x_8)(u)])$

(17) $\exists v[\text{WOMAN}(v) \wedge \forall u[\text{MAN}(u) \rightarrow \text{FIND}(v)(u)]]$

Analogously, the sentence *John seeks a unicorn* is assigned two readings. Formula (18), the *de dicto*-reading, is the outcome of a construction using only (I), (II), and (III). The *de re*-reading, on the other hand, results when the object noun phrase *a unicorn* is quantified into *John seeks* he_3, for instance, using rule (X),3. The translation, (19), reduces to (20).

(18) $\text{SEEK}(^{\wedge}\lambda Q.\exists u[\text{UNICORN}(u) \wedge {}^{\vee}Q(^{\wedge}u)])(j)$

(19) $[\lambda Q.\exists u[\text{UNICORN}(u) \wedge {}^{\vee}Q(^{\wedge}u)]](^{\wedge}\lambda x_3.\text{SEEK}(^{\wedge}\lambda Q.{}^{\vee}Q(x_3))(j))$

(20) $\exists u[\text{UNICORN}(u) \wedge \text{SEEK}(^{\wedge}\lambda Q.{}^{\vee}Q(^{\wedge}u))(j)]$

Notice that the mechanism of quantifying-in creates a *derivational* syntactic ambiguity. For example, whereas the sentence *John seeks a unicorn* is assigned one and the same syntactic *structure* under both of its interpretations, viz.,

(21) $[_{\text{S}}[_{\text{T}}\textit{John}][_{\text{IV}}[_{\text{TV}}\textit{seeks}][_{\text{T}}[_{\text{Det}}\textit{a}][_{\text{CN}}\textit{unicorn}]]]]$,

in our sample derivations of the *de dicto*-reading and the *de re*-reading this structure is *derived* from the two parts

(22) $[_{\text{T}}\textit{John}]$ and $[_{\text{IV}}[_{\text{TV}}\textit{seeks}][_{\text{T}}[_{\text{Det}}\textit{a}][_{\text{CN}}\textit{unicorn}]]]$,

and the two 'parts'

(23) $[_{\text{T}}[_{\text{Det}}\textit{a}][_{\text{CN}}\textit{unicorn}]]$ and $[_{\text{S}}[_{\text{T}}\textit{John}][_{\text{IV}}[_{\text{TV}}\textit{seeks}][_{\text{T}}\textit{he}_3]]]$,

respectively. Here we see that two syntactic derivations may lead to one and the same syntactic structure. The same holds for *Every man finds a woman* on both of its readings.

1.5 Scope Ambiguities: Quantifier Storage

Robin Cooper (1975; 1983) has suggested an alternative approach to quantifier scope ambiguities which has become known as 'Cooper storage'.[7] His treatment is motivated by the idea that 'there is no reason from a syntactic point of view to give [...] more than one syntactic structure'[8] to a sentence like *Every man finds a woman*, depending on which of the quantifiers has wider scope: the rules of quantifying-in create unnecessary (derivational) syntactic ambiguity.

Storage, on the other hand, is a technique of semantic interpretation by means of which scope ambiguities can be obtained without such special syntactic rules. Instead of syntactically quantifying in an NP, one takes the interpretation of the NP and raises it up to the required level in the semantics, where it is assigned its scope. Intuitively,

> if we think of the structural description as a tree and the semantics as working from the bottom of the tree (where the words are) to the top, the storage technique involves putting an NP interpretation on ice for a while until you have interpreted enough of the tree to represent the scope you want to give to the NP. At that point you may take the NP-interpretation out of storage and quantify it in. (Cooper (1983), p. 55)

Technically, the mechanism involves a complication of the process of semantic interpretation. First, syntactic derivations of expressions of English are not assigned a single interpretation, but a *set* of interpretations. And second, these interpretations are themselves ordered sequences (*i*) of which the first member is – a sequence consisting of – a more or less[9] 'traditional' interpretation (viz., an **IL** translation), also called the *denotation* in the terminology of Cooper (1983)); and (*ii*) of which the further members constitute the *store*: a (possibly empty) list of ordered pairs consisting of an

[7] Cooper storage is based on the syntactic technique of 'viable indexing' introduced by Terence Parsons, cf. Cooper and Parsons (1976).

[8] Cooper (1983), p. 3. Cooper, who works within what one could describe as 'transformational Montague grammar', employs the phrase 'syntactic structure' here in the sense of what was called 'syntactic derivation' in the previous section.

[9] I.e., 'more' if the store is empty (then the denotation is a translation familiar from PTQ), and 'less' if the store is filled (then a translation with free variables constitutes the denotation).

NP-denotation δ' together with an indexed variable x_i.[10] In the sequel the variables X and Y will be used to range over (possibly empty) lists consisting of such ordered pairs $\langle \delta', x_i \rangle$ separated by commas.

Working this out for a small fragment which enables us to generate and interpret the sentences treated in Section 1.4, we start with the following lexical interpretation sets:

(24)
$[_{\mathrm{CN}}\textit{man}]$ $\{ \langle\langle \lambda x.[\mathrm{MAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)] \rangle\rangle \}$
$[_{\mathrm{CN}}\textit{woman}]$ $\{ \langle\langle \lambda x.[\mathrm{WOMAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)] \rangle\rangle \}$
$[_{\mathrm{CN}}\textit{unicorn}]$ $\{ \langle\langle \lambda x.[\mathrm{UNICORN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)] \rangle\rangle \}$
$[_{\mathrm{T}}\textit{John}]$ $\{ \langle\langle \lambda P.{}^{\vee}P({}^{\wedge}j) \rangle\rangle \}$
$[_{\mathrm{Det}}\textit{every}]$ $\{ \langle\langle \lambda P \lambda Q.\forall x[{}^{\vee}P(x) \rightarrow {}^{\vee}Q(x)] \rangle\rangle \}$
$[_{\mathrm{Det}}a]$ $\{ \langle\langle \lambda P \lambda Q.\exists x[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)] \rangle\rangle \}$
$[_{\mathrm{TV}}\textit{seeks}]$ $\{ \langle\langle \lambda T \lambda x.\mathrm{SEEK}(T)({}^{\vee}x) \rangle\rangle \}$
$[_{\mathrm{TV}}\textit{finds}]$ $\{ \langle\langle \lambda T \lambda x.{}^{\vee}T({}^{\wedge}\lambda y.\mathrm{FIND}({}^{\vee}y)({}^{\vee}x)) \rangle\rangle \}$

Notice that these interpretation sets are all singleton sets, and that their sole member has a denotation which is identical to the translation assigned in PTQ, while its store is empty.

A crucial rule in the fragment is the rule of *NP-storage* in (25). Let Δ' denote the set of translations of a noun phrase $[_{\mathrm{T}}\delta]$.

(25) If $\langle\langle \delta' \rangle\rangle \in \Delta'$ and $i \in \mathbb{N}$, then $\langle\langle \lambda P.{}^{\vee}P(x_i) \rangle\rangle, \langle \delta', x_i \rangle\rangle \in \Delta'$.

The rule in (25)[11] is intended to supply additional interpretations to the interpretation set of a noun phrase $[_{\mathrm{T}}\delta]$. Thus, in addition to every interpretation $\langle\langle \delta' \rangle\rangle$ that $[_{\mathrm{T}}\delta]$ receives from the lexicon (in case $[_{\mathrm{T}}\delta]$ is lexical) or from the rule that builds up $[_{\mathrm{T}}\delta]$ (in case $[_{\mathrm{T}}\delta]$ is compound), $[_{\mathrm{T}}\delta]$ has an interpretation $\langle\langle \lambda P.{}^{\vee}P(x_i) \rangle\rangle, \langle \delta', x_i \rangle\rangle$ for every natural number i. Formally defined (cf. Section 4 below): Let Δ be the – lexically or syntactically

[10] In Cooper (1983), where the level of IL translations is skipped, the items stored are so-called *binding operators*. The difference is inessential, however, since binding operators are built up from an NP-denotation and an index.

[11] This is the rule of Cooper (1983), p. 61. In Cooper (1975), pp. 192–193, NP-storage is also allowed if the store is non-empty:

If $\langle\langle \delta' \rangle, X \rangle \in \Delta'$ and $i \in \mathbb{N}$, then $\langle\langle \lambda P.{}^{\vee}P(x_i) \rangle\rangle, \langle \delta', x_i \rangle, X \rangle \in \Delta'$.

Condition: i does not occur otherwise as a subscript in the sequence.

The restriction to empty stores in (25) is probably due to the observation that the possibility of storing an NP, part of which is already stored (a possibility which is not ruled out in 1975-style NP-storage), 'would in fact have disastrous consequences in the system proposed in Cooper (1975)' (Cooper (1979), p. 158). Cf. Cooper (1979) and footnote 15 below for further discussion.

assigned – *basic translation set* of $[_{T}\delta]$.[12] Then Δ', the *translation set* of $[_{T}\delta]$, is the smallest set which (*i*) includes Δ; and (*ii*) meets the condition specified in (25).

The rule of *NP-retrieval* has a similar status. That is: let A be the basic translation set of a sentence $[_{S}\alpha]$. Then A′, the translation set of $[_{S}\alpha]$, is the smallest set which (*i*) includes A; and (*ii*) meets the condition specified in (26):[13]

(26) If $\langle\langle\alpha'\rangle, X, \langle\delta', x_i\rangle, Y\rangle \in \mathrm{A}'$, then $\langle\langle\delta'(\hat{}\lambda x_i.\alpha')\rangle, X, Y\rangle \in \mathrm{A}'$.

The syntactic rules which build up compound expressions are listed in (27). Here A′ and B′ denote the translation set of α and β, respectively:

(27)

	RULES OF APPLICATION	BASIC TRANSLATION SET
(I)	$[_{T}\alpha] + [_{IV}\beta] \rightarrow [_{S}[_{T}\alpha][_{IV}\beta]]$	Γ
(II)	$[_{TV}\alpha] + [_{T}\beta] \rightarrow [_{IV}[_{TV}\alpha][_{T}\beta]]$	Γ
(III)	$[_{Det}\alpha] + [_{CN}\beta] \rightarrow [_{T}[_{Det}\alpha][_{CN}\beta]]$	Γ, where

$\Gamma = \{\langle\langle\alpha'(\hat{}\beta')\rangle, X, Y\rangle \mid \langle\langle\alpha'\rangle, X\rangle \in \mathrm{A}' \text{ and } \langle\langle\beta'\rangle, Y\rangle \in \mathrm{B}'\}$.[14]
Condition: if x_i occurs freely in α', then for no δ': $\langle\delta', x_i\rangle \in Y$.[15]

---

[12] The basic translation sets of the present fragment are defined in (24) and (27).

[13] This rule is called *S-quantification* in Cooper (1975), where also *CN-quantification* and *IV-quantification* are formulated: let B′ and Γ′ denote the translation set of a common noun $[_{CN}\beta]$ and an intransitive verb phrase $[_{IV}\gamma]$, respectively:
If $\langle\langle\beta'\rangle, X, \langle\delta', x_i\rangle, Y\rangle \in \mathrm{B}'$, then $\langle\langle\lambda y.\delta'(\hat{}\lambda x_i.\beta'(y))\rangle, X, Y\rangle \in \mathrm{B}'$;
If $\langle\langle\gamma'\rangle, X, \langle\delta', x_i\rangle, Y\rangle \in \Gamma'$, then $\langle\langle\lambda y.\delta'(\hat{}\lambda x_i.\gamma'(y))\rangle, X, Y\rangle \in \Gamma'$.

[14] According to the 'most general definition' given in Cooper (1983), pp. 64–65. In fact, due to peculiarities of the fragment (TVs and Dets happen to always have empty stores), the semantics of (II) and (III) (but not that of (I)) can be simplified as follows:
$\Gamma = \{\langle\langle\alpha'(\hat{}\beta')\rangle, Y\rangle \mid \langle\langle\alpha'\rangle\rangle \in A' \text{ and } \langle\langle\beta'\rangle, Y\rangle \in \mathrm{B}'\}$.

[15] This is in effect the condition given in Cooper (1983), and 'the aim of this restriction is to prevent backwards binding of pronouns', p. 64. For those interested in its history: the condition collapses the two clauses of the *Variable adjustment convention*, which in Cooper (1975) 'prevents certain cases of vacuous quantification by its first clause and [...] backwards pronominalization by its second clause' (p. 193) and which we rephrase here as a condition: if rule R combines expressions α and β with respective translations $\langle\langle\alpha'\rangle, X\rangle$ and $\langle\langle\beta'\rangle, Y\rangle$, then (*i*) for no i, δ', δ'': both $\langle\delta', x_i\rangle \in X$ and $\langle\delta'', x_i\rangle \in Y$; (*ii*) if α occurs to the left of β in the result of applying R, then for no i, δ', δ'': both $\langle\delta', x_i\rangle \in Y$ and x_i occurs freely in $\langle\langle\alpha'\rangle, X\rangle$ but not as $\langle\delta'', x_i\rangle$.

In the present fragment, where all rules R which combine two expressions put the first to the left of the second, this condition is equivalent to: if rule R combines expressions α and β with respective translations $\langle\langle\alpha'\rangle, X\rangle$ and $\langle\langle\beta'\rangle, Y\rangle$ and α occurs to the left of β in the result of applying R, then for no i, δ': both $\langle\delta', x_i\rangle \in Y$ and x_i occurs freely in

Let us now consider the example sentences *John seeks a unicorn* and *Every man finds a woman*. The basic translation sets of their lexical elements were given in (24). The only lexical expression the translation set of which includes non-basic members is the noun phrase *John*, due to NP-storage (25). These non-basic translations constitute the following set:

(28) $\{ \langle\langle \lambda P.{}^{\vee}P(x_i)\rangle, \langle \lambda P.{}^{\vee}P({}^{\wedge}j), x_i\rangle\rangle \mid i \in \mathbb{N} \}$

The compound noun phrases *a unicorn*, *every man* and *a woman* get their respective basic interpretations assigned by application rule (27) (III):[16]

(29) $\{ \langle\langle \lambda Q.\exists u[\text{UNICORN}(u) \wedge {}^{\vee}Q({}^{\wedge}u)]\rangle\rangle \}$,
$\{ \langle\langle \lambda Q.\forall u[\text{MAN}(u) \rightarrow {}^{\vee}Q({}^{\wedge}u)]\rangle\rangle \}$, and
$\{ \langle\langle \lambda Q.\exists v[\text{WOMAN}(v) \wedge {}^{\vee}Q({}^{\wedge}v)]\rangle\rangle \}$.

Again, NP-storage (25) supplies an infinite number of non-basic translations:

(30) $\{ \langle\langle \lambda P.{}^{\vee}P(x_i)\rangle, \langle \lambda Q.\exists u[\text{UNICORN}(u) \wedge {}^{\vee}Q({}^{\wedge}u)], x_i\rangle\rangle \mid i \in \mathbb{N} \}$,
$\{ \langle\langle \lambda P.{}^{\vee}P(x_i)\rangle, \langle \lambda Q.\forall u[\text{MAN}(u) \rightarrow {}^{\vee}Q({}^{\wedge}u)], x_i\rangle\rangle \mid i \in \mathbb{N} \}$, and
$\{ \langle\langle \lambda P.{}^{\vee}P(x_i)\rangle, \langle \lambda Q.\exists v[\text{WOMAN}(v) \wedge {}^{\vee}Q({}^{\wedge}v)], x_i\rangle\rangle \mid i \in \mathbb{N} \}$,

---

$\langle\langle \alpha'\rangle, X\rangle$. Moreover, 1983-style NP-storage (25) has the corollary that (modulo pronouns) no x_i can occur freely in X without doing so in α', so in the present pronounless fragment this condition is equivalent to the condition on (27), which requires that for no i, δ' it be the case that both $\langle \delta', x_i\rangle \in Y$ and x_i occurs freely in α'.

Putting aside pronoun binding, note that this condition is also necessary for avoiding vacuous quantification. E.g., without it we could store both $\langle \lambda Q.\forall v {}^{\vee}Q({}^{\wedge}v), x_5\rangle$ and $\langle \lambda Q.\exists u {}^{\vee}Q({}^{\wedge}u), x_5\rangle$ while interpeting the sentence *Everything loves something*, which would eventually yield the non-interpretations $\exists u \forall v \text{LOVE}(v)(v)$, 'everything loves itself', and $\forall v \exists u \text{LOVE}(u)(u)$, 'something loves itself', depending on the order of NP-retrieval.

In the system of Cooper (1975), however, the condition is not sufficient for the prevention of all cases of vacuous quantification: in the process of interpreting the sentence *Every woman who loves a man walks* we can first store $\langle \lambda Q.\exists v[\text{MAN}(v) \wedge {}^{\vee}Q({}^{\wedge}v)], x_7\rangle$ and then $\langle \lambda Q.\forall u[[\text{WOMAN}(u) \wedge \text{LOVE}({}^{\vee}x_7)(u)] \rightarrow {}^{\vee}Q({}^{\wedge}u)], x_4\rangle$. When these items are retrieved from storage in the same order, the whole sentence will receive the non-interpretation $\forall u[[\text{WOMAN}(u) \wedge \text{LOVE}({}^{\vee}x_7)(u)] \rightarrow \exists v[\text{MAN}(v) \wedge \text{WALK}(u)]]$. This predicament is avoided in Cooper (1983) by allowing NP-storage only if the store is empty. (What looks like a straightforward alternative way out – viz., requiring that the translations of α and β have no shared free variables at all – would in effect result in an undesirable universal ban on pronoun binding.)

[16] We will assume that λ-conversion, ${}^{\vee}{}^{\wedge}$-elimination and other IL reductions lead to equivalent members of ordered sequences of members of ordered sequences. We will identify member-of-memberwise equivalent sequences of sequences and write down only reduced ones.

respectively. Using application rule (27) (II), one obtains the basic translation sets listed in (31) and (32) for the IVs *seeks a unicorn* and *finds a woman*, respectively. Because NP-storage and NP-retrieval do not apply to the category of intransitive verb phrases, the translation sets of these IVs do not contain additional non-basic members. The storeless members of (31) and (32) stem from the basic translation of the object NPs, whereas for the remaining ones the non-basic translations in (30) have been employed.

(31) $\{\ \langle\langle \lambda x.\textsc{seek}(^\wedge\lambda Q.\exists u[\textsc{unicorn}(u) \wedge {}^\vee Q(^\wedge u)])(x)\rangle\rangle\ \} \cup$
$\{\ \langle\langle \lambda x.\textsc{seek}(^\wedge\lambda P.{}^\vee P(x_i))({}^\vee x)\rangle\rangle,$
$\langle \lambda Q.\exists u[\textsc{unicorn}(u) \wedge {}^\vee Q(^\wedge u)], x_i\rangle\rangle \mid i \in \mathbb{N}\ \}$

(32) $\{\ \langle\langle \lambda x.\exists v[\textsc{woman}(v) \wedge \textsc{find}(v)({}^\vee x)]\rangle\ \} \cup$
$\{\ \langle\langle \lambda x.\textsc{find}({}^\vee x_i)({}^\vee x)\rangle\rangle,$
$\langle \lambda Q.\exists v[\textsc{woman}(v) \wedge {}^\vee Q(^\wedge v)], x_i\rangle\rangle \mid i \in \mathbb{N}\ \}$

Application rule (27) (I) assigns the basic translation sets (33) and (34) to the sentences *John seeks a unicorn* and *Every man finds a woman*.

(33) $\{\ \langle\langle \textsc{seek}(^\wedge\lambda Q.\exists u[\textsc{unicorn}(u) \wedge {}^\vee Q(^\wedge u)])(j)\rangle\rangle\ \}^{\{1\}} \cup$
$\{\ \langle\langle \textsc{seek}(^\wedge\lambda Q.\exists u[\textsc{unicorn}(u) \wedge {}^\vee Q(^\wedge u)])({}^\vee x_i)\rangle\rangle,$
$\langle \lambda P.{}^\vee P(^\wedge j), x_i\rangle\rangle \mid i \in \mathbb{N}\ \}^{\{2\}} \cup$
$\{\ \langle\langle \textsc{seek}(^\wedge\lambda P.{}^\vee P(x_i))(j)\rangle\rangle,$
$\langle \lambda Q.\exists u[\textsc{unicorn}(u) \wedge {}^\vee Q(^\wedge u)], x_i\rangle\rangle \mid i \in \mathbb{N}\ \}^{\{3\}} \cup$
$\{\ \langle\langle \textsc{seek}(^\wedge\lambda P.{}^\vee P(x_j))({}^\vee x_i)\rangle\rangle,$
$\langle \lambda P.{}^\vee P(^\wedge j), x_i\rangle,$
$\langle \lambda Q.\exists u[\textsc{unicorn}(u) \wedge {}^\vee Q(^\wedge u)], x_j\rangle\rangle \mid i, j \in \mathbb{N} \text{ and } i \neq j\ \}^{\{4\}}$

(34) $\{\ \langle\langle \forall u[\textsc{man}(u) \rightarrow \exists v[\textsc{woman}(v) \wedge \textsc{find}(v)(u)]]\rangle\rangle\ \}^{\{1\}} \cup$
$\{\ \langle\langle \exists v[\textsc{woman}(v) \wedge \textsc{find}(v)({}^\vee x_i)]\rangle,$
$\langle \lambda Q.\forall u[\textsc{man}(u) \wedge {}^\vee Q(^\wedge u)], x_i\rangle\rangle \mid i \in \mathbb{N}\ \}^{\{2\}} \cup$
$\{\ \langle\langle \forall u[\textsc{man}(u) \rightarrow \textsc{find}({}^\vee x_i)(u)]\rangle,$
$\langle \lambda Q.\exists v[\textsc{woman}(v) \wedge {}^\vee Q(^\wedge v)], x_i\rangle\rangle \mid i \in \mathbb{N}\ \}^{\{3\}} \cup$
$\{\ \langle\langle \textsc{find}({}^\vee x_j)({}^\vee x_i)\rangle\rangle,$
$\langle \lambda Q.\forall u[\textsc{man}(u) \rightarrow {}^\vee Q(^\wedge u)], x_i\rangle,$
$\langle \lambda Q.\exists v[\textsc{woman}(v) \wedge {}^\vee Q(^\wedge v)], x_j\rangle\rangle \mid i, j \in \mathbb{N} \text{ and } i \neq j\ \}^{\{4\}}$

The storeless translations of the subject NPs in (24) and (29) are responsible for {1} and {3} of (33) and (34), and the 'store-keeping' ones in (28) and

(30) lead to {2} and {4}. For {1} and {2} of (33) and (34), the first members of the translation sets (31) and (32) have been used, while the remaining ones have contributed to {3} and {4}. The requirement that $i \neq j$ in (33) and (34) {4} is a consequence of the *Condition* on (27): a free variable in the denotation of the subject NP is not allowed to occur in the store of the interpretation of the VP.

Due to NP-retrieval (26), the sentences also have non-basic translations: translations which are not equivalent to the ones in (33) and (34). NP-retrieval is applicable to {2}, {3} and {4} of (33) and (34). When it is applied to some member of {2}, nothing new arises: the resulting translation is equivalent to the only member of {1}. The same holds when NP-retrieval is applied to some member of {4}. If the first NP in the store is retrieved, a member of {2} will be obtained; if the last NP in the store is retrieved, a member of {3} will come out. More interesting things happen when NP-retrieval is applied to members of {3}. This yields one new storeless translation, both for *John seeks a unicorn* and for *Every man finds a woman*:

(35) $\langle\langle \exists u[\text{UNICORN}(u) \wedge \text{SEEK}(\hat{}\lambda P.\check{}P(\hat{}u))(j)]\rangle\rangle$ and

(36) $\langle\langle \exists v[\text{WOMAN}(v) \wedge \forall u[\text{MAN}(u) \rightarrow \text{FIND}(v)(u)]]\rangle\rangle$.

In sum, we see that among the infinite number of translations assigned to each of the sentences there are two storeless ones, i.e., translations of the form $\langle\langle\alpha\rangle\rangle$, viz., (33) {1} and (35), and (34) {1} and (36).

According to Cooper, 'it is only in this case that we have a bona fide [...] denotation' (Cooper (1983), p. 56), for since 'we identify singleton sequences with their members',[17] 'any member of a translation set which is not complex may be thought of as a meaning of that expression'.[18]

[17]Cooper (1975), p. 153. There he adds: 'This adjusted formulation allows us to preserve the notion that the translation of each successively larger part of the tree is a function of the translation of its immediately smaller parts.' Thus, Cooper seems to claim that his grammar is compositional. However, this cannot be meant to apply in the ordinary way, i.e., that in Cooper's grammar, bona fide denotations (i.e., singleton sequences consisting of a singleton sequence consisting of an expression of **IL**) of each succesively larger part of the tree are a function of bona fide denotations of its immediately smaller parts, because this is obviously false. This is not to say that the grammar is not compositional at all: it is (or can easily be made) compositional – albeit on the level of *sequences of sequences of* meanings. (Cf. the next footnote.)

[18]Cooper even goes on to say that 'those members of the translation set which are

Weighing the additional complexities of semantic interpretation against the simpler ambiguous syntax in his treatment of scope ambiguities, Cooper draws the following conclusion:

> What seems unappetizing about this system is not that we map into sets of meanings but that we have to map into sets of *sequences of sequences* of meanings. (Cooper (1975), p. 160)

In Section 4 we will develop a grammar for scope ambiguities which shares the first property with Cooper's system, but lacks the second one.

2 Arguments for Flexibility

We saw in Section 1.2 above that, due to the rigid category-to-type assignment in PTQ, a proper name like *John* cannot be translated as an individual constant of type e, and is – perforce, so to speak – assigned $\lambda P.^{\vee}P(^{\wedge}j)$, a complex expression of **IL** of type $((s,((s,e),t)),t)$, as its translation.

This, in its turn, has further consequences: an extensional transitive verb like *finds*, for example, is translated as $\lambda T\lambda x.^{\vee}T(^{\wedge}\lambda y.\text{FIND}(^{\vee}y)(^{\vee}x))$ of type $\text{TYPE}(\text{TV}) =$

$$((s,((s,((s,e),t)),t)),((s,e),t));$$

and not as $\lambda y\lambda x.\text{FIND}(^{\vee}y)(^{\vee}x)$ of type

$$((s,e),((s,e),t)),$$

which would be the most direct way to obtain the translation $\text{FIND}(m)(j)$ for *John finds Mary* under the assumptions that proper names have denotations of type e, and that the semantics of the rules (I) and (II) is intensional functional application.

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 the subject position of which is intensional, cf. (37) and (38):

(37) A man is missing

---

complex do not represent meanings but rather sequences thereof. We may account for the fact that speakers of English do not utter sentences which are assigned complex sequences of meanings with [a] pragmatic maxim [...]. It is impossible to determine a truth-value relative to a sequence of meanings and therefore a sentence which is assigned a complex sequence is not information bearing.' (Cooper (1975), pp. 153–154)

(38) A unicorn seems to be approaching

In BE-MISSING($^{\wedge}\lambda Q.\exists v[\text{MAN}(v) \wedge {}^{\vee}Q(v)]$), an **IL** rendering of the non-referential reading of (37) which does not entail the existence of men, the translation of the intransitive verb phrase, BE-MISSING, is not of type TYPE(IV) – i.e., type $((s,e),t)$ – but of type $((s,\text{TYPE(T)}),t)$ =

$$((s,((s,((s,e),t)),t)),t).$$

Mutatis mutandis, the same holds for sentence (38).

The existence of intransitive verb phrases like *is missing* entails a raising of the type assignment (plus a corresponding change in the categorial definition) of a number of syntactic categories within the PTQ fragment: for *all* intransitive verbs (hence also for extensional ones like *walks*) 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):

$$((s,((s,((s,e),t)),t)),((s,((s,((s,e),t)),t)),t));^{19}$$

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 (1970b)), extended with 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 – cannot be pursued successfully. This has been claimed by Rooth and Partee (1982), Partee and Rooth (1983), Groenendijk and Stokhof (1984; 1989), and Partee (1986), among others.

The starting point of Partee and Rooth (1983) is a generalized, 'cross-categorial' semantics for coordination by means of *and* and *or* (see also Von

[19] While no transitive verb with this amount of intensionality is known; cf. Partee and Rooth's claim that 'there are [apparently] no basic lexical verbs that are intensional with respect to both subject and object' (Partee and Rooth (1983), p. 379).

Stechow (1974), Gazdar (1980), Keenan and Faltz (1985), Groenendijk and Stokhof (1989)). This semantics is based on the set CT of 'conjoinable types', which is defined as the smallest set satisfying the requirements that (*i*) $t \in CT$; and (*ii*) if $b \in CT$, then $(a, b) \in CT$.

Notice that every conjoinable type is of the form $(a_1, (\ldots (a_n, t) \ldots))$, where $a_1, \ldots, a_n$ are arbitrary types and $n \geq 0$. For b a type and $\vec{a}$ a (possibly empty) sequence of types $a_1, \ldots, a_n$, we will let $(\vec{a}, b)$ represent the type $(a_1, (\ldots (a_n, b) \ldots))$. Moreover, if $\vec{a}$ is such a sequence of types, then $\vec{x}_{\vec{a}}$ denotes the sequence of variables $x_1, \ldots, x_n = v_{1,a_1}, \ldots, v_{n,a_n}$ (i.e., for all i such that $1 \leq i \leq n$: x_i is the i-th variable of type a_i); $\lambda \vec{x}_{\vec{a}}.\tau$ denotes $\lambda v_{1,a_1} \ldots \lambda v_{n,a_n}.\tau$; and $\tau(\vec{x}_{\vec{a}})$ denotes $\tau(v_{1,a_1}) \ldots (v_{n,a_n})$.

Given this much, generalized conjunction and disjunction can be defined as in (39) and (40):[20]

(39) GC (Generalized Conjunction):
for γ and δ of type $(\vec{a}, t)$: $\mathrm{GC}(\gamma, \delta) = \lambda \vec{x}_{\vec{a}}.[\gamma(\vec{x}) \wedge \delta(\vec{x})]$

(40) GD (Generalized Disjunction):
for γ and δ of type $(\vec{a}, t)$: $\mathrm{GD}(\gamma, \delta) = \lambda \vec{x}_{\vec{a}}.[\gamma(\vec{x}) \vee \delta(\vec{x})]$

Note that for transitive verb phrases $[_{\mathrm{TV}}\ [_{\mathrm{TV}} \alpha]$ *and* $[_{\mathrm{TV}} \beta]]$, this generalized semantics entails the interpretation given in (41) if the translations α' and β' of $[_{\mathrm{TV}} \alpha]$ and $[_{\mathrm{TV}} \beta]$ are of type $((s, e), ((s, e), t))$, and the interpretation given in (42) if α' and β' are of type $((s, ((s, ((s, e), t)), t)), ((s, e), t))$:

(41) $\lambda x_{(s,e)} \lambda y_{(s,e)}.[\alpha'(x)(y) \wedge \beta'(x)(y)]$

(42) $\lambda T_{(s,((s,((s,e),t)),t))} \lambda y_{(s,e)}.[\alpha'(T)(y) \wedge \beta'(T)(y)]$

Next, Partee and Rooth observe that if extensional verbs like *caught* and *ate* are uniformly assigned the PTQ interpretations of the higher-order type

[20] In fact, these definitions are only correct if the variables in $\vec{x}_{\vec{a}}$ do not occur freely in γ and δ: if they do, 'accidental binding' will take place (cf. footnote 5 above). In order to let (39) and (40) work for *any* $\gamma_{(\vec{a},t)}$ and $\delta_{(\vec{a},t)}$ (the same holds for (59), (60) and (61) below, and for the operations of value raising, argument raising and argument lowering to be defined in Section 4.2), there are at least three options.

The first option is chosen in PTQ (cf. footnote 10 above), where the free variables in translations and the variables of binding operators in semantic rules which should not bind those free variables constitute two disjoint sets: the variables with even and odd subscripts, respectively. Under this approach (39) and (40) can remain unchanged, but $\vec{x}_{\vec{a}}$ must be redefined as $v_{2\cdot 1+1,a_1}, \ldots, v_{2\cdot n+1,a_n}$ (i.e., for all i such that $1 \leq i \leq n$: x_i is the $(2 \cdot i + 1)$-th variable of type a_i).

$((s,((s,((s,e),t)),t)),((s,e),t))$ in (43),[21] then instance (42) of generalized conjunction (39) will predict that sentence (44) gets the interpretation given in (45):

(43) $\lambda T\lambda y.{}^{\vee}T({}^{\wedge}\lambda x.\text{CATCH}({}^{\vee}x)({}^{\vee}y))$ and $\lambda T\lambda y.{}^{\vee}T({}^{\wedge}\lambda x.\text{EAT}({}^{\vee}x)({}^{\vee}y))$

(44) John caught and ate a fish

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

Partee and Rooth do not accept (45) as a possible interpretation of (44). The interpretation which they do assign to (44) is (46):

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

Alternatively, one can generalize Janssen's correction of Thomason's revision of Montague's semantics $\lambda x_i.[\alpha'(x_i) \wedge \beta']$ for the rule (scheme) $S_3, i/T_3, i$ in PTQ (Janssen (1986), Part 2, p. 159; Thomason (ed.) (1974), p. 261). In order to avoid the problem of 'collision of variables' – the undesirable possibility that free occurrences of x_i in α' become bound by $S_3, i/T_3, i$ – Janssen proposes $[\lambda P\lambda x_i.[{}^{\vee}P(x_i) \wedge \beta']]({}^{\wedge}\alpha')$, a solution which exploits the conditions on λ-conversion in **IL**: the expression ${}^{\wedge}\alpha'$ is intensionally closed, so if x_i does not occur freely in α', then the formula $[\lambda P\lambda x_i.[{}^{\vee}P(x_i) \wedge \beta']]({}^{\wedge}\alpha')$ is equivalent to $[\lambda x_i.[{}^{\vee}{}^{\wedge}\alpha'(x_i) \wedge \beta']]$, and hence to $[\lambda x_i.[\alpha'(x_i) \wedge \beta']]$. More generally, if τ is a translation rule with metavariables $\alpha_1, \ldots, \alpha_n$ of types $a_1, \ldots, a_n$ occurring in the scope of a variable binding operator which should not bind free variable occurrences in $\alpha_1, \ldots, \alpha_n$, then replace τ by

$$[\lambda x_1 \ldots \lambda x_n.[\tau[\alpha_1 := {}^{\vee}x_1, \ldots, \alpha_n := {}^{\vee}x_n]]]({}^{\wedge}\alpha_1)\ldots({}^{\wedge}\alpha_n),$$

where $x_1, \ldots, x_n$ are different variables of respective types $(s,a_1), \ldots, (s,a_n)$ which do not occur in τ. This leaves the definition of $\vec{x}_{\vec{a}}$ as it is, but changes (39) into:

$$\text{GC}(\gamma,\delta) = [\lambda v_{n+1,(s,(\vec{a},t))}\lambda v_{n+2,(s,(\vec{a},t))}\lambda\vec{x}_{\vec{a}}.[{}^{\vee}v_{n+1}(\vec{x}) \wedge {}^{\vee}v_{n+2}(\vec{x})]]\,({}^{\wedge}\gamma)({}^{\wedge}\delta).$$

Finally, one can impose a general ban on free variables in translations of expressions. The grammar of Section 4 below satisfies this *reductio ad minimum* of Janssen's 'variable principle' (Janssen (1986), Part 2, pp. 137–173). Moreover, Groenendijk and Stokhof (1990) have shown that this requirement is not inconsistent with a fairly proper treatment of anaphora (see also Section 6 below). They note that 'variables have (at least) two functions. [They] are used in the representation of quantificational expressions, and of anaphoric relations between such expressions and pronouns. [...] The variables occurring [in λ-abstracts] play a different role. In such constructs [...], variables don't have a direct natural language counterpart' (pp. 5–6). Correspondingly, the grammar of Groenendijk and Stokhof distinguishes between ordinary variables (for λ-abstracts), and so-called *discourse markers*, which are used in the representation of quantification and anaphora. (Strictly speaking, they render quantification as a combination of a variable-binding quantifier and a so-called *state switcher*, but this is equivalent to straightforward quantification over discourse markers as employed in Dekker (1990).) The introduction of 'discourse markers and state switchers is really the same as adding a second kind of variable-and-binding device' (Groenendijk and Stokhof (1990), p. 14).

[21] We will slightly adapt the types of Partee and Rooth to the usual PTQ types.

Interpretation (46) can be brought about by letting *caught* and *ate* have the following interpretations of type $((s,e),((s,e),t))$:

(47) $\lambda x\lambda y.\text{CATCH}(^{\vee}x)(^{\vee}y)$ and $\lambda x\lambda y.\text{EAT}(^{\vee}x)(^{\vee}y)$

Instance (41) of generalized conjunction (39) will then interpret the compound transitive verb phrase *caught and ate* as (48):

(48) $\lambda x\lambda y.[\text{CATCH}(^{\vee}x)(^{\vee}y) \wedge \text{EAT}(^{\vee}x)(^{\vee}y)]$

Suppose now that there is a 'redundancy rule' to insure that each transitive verb phrase α with interpretation α' of type $((s,e),((s,e),t))$ has a predictable homonym of type $((s,((s,((s,e),t)),t)),((s,e),t))$, viz.:[22]

(49) $\lambda T_{(s,((s,((s,e),t)),t))}\lambda y_{(s,e)}.{}^{\vee}T(^{\wedge}\lambda x_{(s,e)}.\alpha'(x)(y))$

With this rule, *caught and ate* will be predicted to have the interpretation in (50):

(50) $\lambda T\lambda y.{}^{\vee}T(^{\wedge}\lambda x.[\text{CATCH}(^{\vee}x)(^{\vee}y) \wedge \text{EAT}(^{\vee}x)(^{\vee}y)])$

Finally, combination of (50) with the direct object *a fish* and the subject *John* yields (46). It is important that this 'redundancy rule' is only invoked when it is really necessary (i.e., for making interpretations combinable). For example, if it is freely applicable to the lexical interpretations in (47), then interpretation (45) will be assigned after all.

Partee and Rooth note that if, on the other hand, two intensional verbs (cf. (51)) or an intensional verb and an extensional verb (cf. (53)) are conjoined, then the PTQ-translations of type $((s,((s,((s,e),t)),t)),((s,e),t))$ and generalized conjunction (42) do yield the 'primary [*de dicto*] readings'[23]

[22] Applied to $\lambda x\lambda y.\text{FIND}(^{\vee}x)(^{\vee}y)$, the rule will yield the PTQ-translation of *finds*.

[23] Non-primary *de re*-readings of (51) and (53) ('there is a secretary/coat such that ... and ...') 'may also be available [...], with about as much ease or difficulty as with a single intensional verb' (p. 366). For such readings, Partee and Rooth assume some quantifying-in mechanism (which they do not specify further). Observe that quantifying in *a coat* into the sentence *John needed and bought he$_i$* or into the intransitive verb phrase *needed and bought he$_i$* will only result in a *de re* reading of (53) in which *a coat* takes scope over *and*:

$$\exists v[\text{COAT}(v) \wedge [\text{NEED}(^{\wedge}\lambda P.{}^{\vee}P(^{\wedge}v))(j) \wedge \text{BUY}(v)(j)]]$$

A *de re* reading in which *and* has wider scope than *a coat* cannot be obtained by quantifying in:

$$\exists v[\text{COAT}(v) \wedge \text{NEED}(^{\wedge}\lambda P.{}^{\vee}P(^{\wedge}v))(j)] \wedge \exists v[\text{COAT}(v) \wedge \text{BUY}(v)(j)]$$

Mutatis mutandis, the same holds for sentence (51).

(52) and (54) of the sentences:

(51) John wants and needs a secretary

(52) $\text{WANT}(^\wedge\lambda P.\exists v[\text{SECRETARY}(v) \wedge {}^\vee P(^\wedge v)])(j)$
$\wedge\ \text{NEED}(^\wedge\lambda P.\exists v[\text{SECRETARY}(v) \wedge {}^\vee P(^\wedge v)])(j)$

(53) John needed and bought a coat

(54) $\text{NEED}(^\wedge\lambda P.\exists v[\text{COAT}(v) \wedge {}^\vee P(^\wedge v)])(j)$
$\wedge\ \exists v[\text{COAT}(v) \wedge \text{BUY}(v)(j)]$

Parallel issues are claimed to arise with intransitive verbs. Thus if two extensional intransitive verbs are coordinated as in (55), then the correct interpretation (56) requires their conjunction at the level of type $((s,e),t))$, but if the coordination involves two intensional verbs, or an intensional verb and an extensional verb (cf. (57) and (58), respectively), then the conjunction must take place at the level of type $((s,((s,((s,e),t)),t)),t)$ in order to admit *de dicto*-readings.[24]

(55) A fish walked and talked

(56) $\exists v[\text{FISH}(v) \wedge [\text{WALK}(v) \wedge \text{TALK}(v)]]$

(57) An easy textbook is needed and will surely be written

(58) A guest was expected and did in fact show up

In view of such examples, Partee and Rooth conclude that we should give up Montague's strategy of uniformly assigning *all* members of a certain syntactic category the 'highest' type that is needed for *some* expression in that category. Instead, they formulate the following three-part proposal (pp. 367–368):

[24] Again, a quantifying-in mechanism may take care of possible additional *de re*-readings of (57) and (58) in which the subject noun phrase obtains the widest scope. Sentence (55), however, is claimed to mean only (56), and to *lack* the reading

$$\exists v[\text{FISH}(v) \wedge \text{WALK}(v)] \wedge \exists v[\text{FISH}(v) \wedge \text{TALK}(v)].$$

Partee and Rooth abandon the idea of a 'fixed directionality' to the function-argument structure of subject and predicate: if a verb phrase of type $((s,((s,((s,e),t)),t)),t)$ is combined with a subject noun phrase of type $((s,((s,e),t)),t)$, then (the intension of) the noun phrase is the argument and the verb phrase is the functor. The same holds for a verb phrase of type $((s,e),t)$ and a subject noun phrase of type e.

(*a*) Enter each verb lexically in its minimal type.

(*b*) Provide lexical rules furnishing 'higher'-type homonyms for 'lower'-type elements.

(*c*) Posit as a processing strategy that all expressions are interpreted at the lowest type possible, invoking higher-type homonyms only when needed for type coherence.

As a matter of fact, part (*a*) of the proposal also holds for non-verbal lexical elements in the fragment presented by Partee and Rooth: besides TVs of type $((s,e),((s,e),t))$ and $((s,((s,((s,e),t)),t)),((s,e),t))$ (*find* vs. *seek*) and IVs of type $((s,e),t)$ and $((s,((s,((s,e),t)),t)),t)$ (*walks* vs. *is missing*), there are NPs of type e and $((s,((s,e),t)),t)$ (*John* vs. *someone*).

Part (*b*) of the proposal makes mention of 'lexical rules'. But as we have seen above (cf. (50)), it is crucial that these 'redundancy rules' are allowed to apply to non-lexical expressions as well. Partee and Rooth formulate three redundancy rules. For interpretations α'_e of $[_{\mathrm{T}}\alpha]$, $\beta'_{((s,e),t)}$ of $[_{\mathrm{IV}}\beta]$, and $\gamma_{((s,e),((s,e),t))}$ of $[_{\mathrm{TV}}\gamma]$, they propose the higher-type homonyms (59) of type $((s,((s,e),t)),t)$, (60) of type $((s,((s,((s,e),t)),t)),t)$, and (61) of type $((s,((s,((s,e),t)),t)),((s,e),t))$, respectively:[25]

(59) $\lambda P_{(s,((s,e),t))}.{}^{\vee}P({}^{\wedge}\alpha')$

(60) $\lambda T_{(s,((s,((s,e),t)),t))}.{}^{\vee}T({}^{\wedge}\beta')$

(61) $\lambda T_{(s,((s,((s,e),t)),t))}\lambda y_{(s,e)}.{}^{\vee}T({}^{\wedge}\lambda x_{(s,e)}.\gamma'(x)(y))$

Part (*c*) of Partee and Rooth's alternative for Montague's 'generalizing to the worst case' could be called the 'coercion principle'. For every expression there is a 'best case' and a 'worst case' interpretation (which may coincide – e.g., for determiners, which lack a redundancy rule). The former is the default one, while the latter is only invoked when one is coerced into using it: 'try simplest types first'. So, though there is still a worst case, one cannot generalize to it, on account of the coercion principle.

Groenendijk and Stokhof (1984; 1989) argue that type-shifting rules are a useful tool in the semantic description of various constructions involving interrogatives. They show that an extension of the rules (59) and (61) to

[25] Rule (61) repeats (49).

other categories (and types) allows for a representation of desired readings, and, furthermore, that certain intuitive entailments between (coordinated) interrogatives[26] – a conjunction entails its conjuncts; a disjunction is entailed by its disjuncts – can only be accounted for if it is assumed that rule (59) admits of interpretations of interrogatives in a higher type than their basic type (s,t). On the other hand, type (s,t) is needed as well: for entailments between atomic interrogatives. Accordingly, the three-part proposal presented by Groenendijk and Stokhof (1989, p. 60) differs from the one offered by Partee and Rooth. In particular, the coercion principle is replaced by the following:

> Interpretation of syntactic structures is liberalized to a relation: 'anything goes that fits'. I.e., a syntactic structure has as many meanings as can be generated from the potential meanings of its constituents.[27]

As Partee and Rooth note, their coercion principle precludes a use of type-shifting for the representation of ambiguities. They discuss an interesting example which is structurally identical to the following one:

(62) John seeks a fish or a bike

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

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

Sentence (62) has the *de dicto* reading (63) and the *de re* reading (64) to start with. The *de dicto* reading can be accounted for in PTQ by applying rule (IX) to *a fish* and *a bike*, building up *a fish or a bike* with the reduced translation (65):

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

If we combine (65) with the translations of *seek* (rule (II)) and *John* (rule (I)) in that order, then the result reduces to (63). Reading (64) is also

[26] Using a generalized notion of entailment along the lines of (39) and (40), according to which an interrogative A entails an interrogative B just in case every complete and true answer to A is a complete and true answer to B.

[27] In Section 4 below we will implicitly employ a similar principle.

among the possibilities of PTQ. For this reading we first build up *seek* he_i (with rule (II)), for some $i \in \mathbb{N}$. Rule (I) combines this intransitive verb phrase with the term *John*. The resulting 'sentence' is *John seeks* he_i. With some rule (X),i we then quantify in the disjunctive noun phrase *a fish or a bike* into *John seeks* he_i:

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

Expression (66) is equivalent to (64).

But (63) and (64) do not exhaust the possible interpretations of (62). As Partee and Rooth observe, 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'. It can be formulated in **IL** as:

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

PTQ does not account for (67). Partee and Rooth indicate how this reading can be constructed. Suppose that every expression β with translation β' of type b also has a translation of type $((s,((s,b),c)),c)$:[28]

(68) $\lambda Q_{(s,((s,b),c))}.{}^{\vee}Q(^{\wedge}\beta')$,

then (with $c = ((s,e),t)$) the noun phrases *a fish* and *a bike* have translations – see (70) – of the following type:

(69) $((s,((s,((s,((s,e),t)),t)),((s,e),t))),((s,e),t))$

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

Suppose, in addition, that the combination of these expressions into *a fish or a bike* follows the general syntactic/semantic scheme of generalized disjunction defined in (40), then the compound noun phrase *a fish or a bike* is assigned translation (71), which is also of type (69):

(71) $\lambda Q\lambda y.[{}^{\vee}Q(^{\wedge}\lambda P.\exists v[\text{FISH}(v) \wedge {}^{\vee}P(^{\wedge}v)])(y)$
$\vee\, {}^{\vee}Q(^{\wedge}\lambda P.\exists v[\text{BIKE}(v) \wedge {}^{\vee}P(^{\wedge}v)])(y)]$

[28] Rule (68) is a threefold generalization of (59) (and of (60)): (*i*) it applies to translations of expressions of any category; (*ii*) it applies to translations of any type b; and (*iii*) it yields translations of type $((s,((s,b),c)),c)$ for any type c.

This step violates the coercion principle, since the type coherence needed for generalized disjunction – identity – is already displayed by the translations obtained by combining the lexical translations of *a* and *fish*, and *a* and *bike*.[29]

Application of (71) to the intension of $\lambda T \lambda x.\text{SEEK}(T)(^{\vee}x)$ reduces to (72). And application of (72) to the intension of j yields (67).

(72) $\lambda y.[\text{SEEK}(^{\wedge}\lambda P.\exists v[\text{FISH}(v) \wedge {}^{\vee}P(^{\wedge}v)])(^{\vee}y)$
$\vee\ \text{SEEK}(^{\wedge}\lambda P.\exists v[\text{BIKE}(v) \wedge {}^{\vee}P(^{\wedge}v)])(^{\vee}y)]$

The purpose of the coercion principle is the prevention of undesired readings such as (73) for sentence (74).

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

(74) John caught and ate a fish

Partee and Rooth insist that (74) must be interpreted as involving just one fish, 'unless the sentence [is] given a very marked intonation or the context is heavily loaded' (p. 365). On the other hand, we think that this observation is at least questionable, in consideration of the natural continuation[30] indicated in (75):

(75) John caught and ate a fish. The fish he caught proved to be inedible, whereupon the fish he ate caught his eye.

---

[29]In Rooth and Partee (1982), an alternative treatment of wide scope-*or* along the lines of Kamp (1981) and Heim (1982) is proposed: there, the disjunction introduces a free variable (discourse marker) together with a disjunctive condition on that variable. The authors conclude, however, that 'considerable work remains to be done to turn these sketchy remarks into an explicit set of rules'.

[30]That is to say, extrapolating from the intuitions of a dozen of Dutch speakers and a couple of German ones. All of them considered (the Dutch/German counterpart of) sentence (74) true in a situation where John caught and ate only one fish, viz., a goldfish and a kipper. Moreover, the possibility of readings like (73) seems to be independent of context and intonation. Someone could very well start a conversation with sentence (*i*) while referring to two different cars. And someone else could react to that using any of the sentences in (*ii*):

(*i*) Yesterday John bought and sold a car

(*ii*) No, *today* John bought and sold a car
No, yesterday *Mary* bought and sold a car
No, yesterday John *rented* and sold a car
No, ...

In cases involving intransitive verbs (*A fish walked and talked*) it is harder to get a reading where the conjunction takes scope over the subject quantifier, but still not impossible – given some 'loading' of the context: [31]

(76) A man invented dynamite and devised the neutron bomb, but a woman discovered X rays.

In view of the above considerations we will give up the coercion principle. This liberalization will enable us to account for ambiguities by means of type-shifting principles. Montague's strategy of 'generalizing to the worst case' will fail in the resulting grammar, though for a different reason than in the grammar of Partee and Rooth. It fails, not because the worst case cannot always be generalized to, but simply because there *is* no such case.

That this is in correspondence with the facts can be shown as follows. We have seen that in order to obtain the *de dicto* wide scope-*or* reading (67) of the sentence *John seeks a fish or a bike* by means of rule (68), Partee and Rooth need type (69)[32] – that is: $((s, \text{TYPE}(\text{TV})), ((s, e), t))$ – for the noun phrases *a fish* and *a bike*.

(77) 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

Observe that the sentences in (77) all have readings which express an uncertainty of the speaker. These *de dicto* wide scope-*or* readings are suggested by the continuation '... but I don't know exactly what she claims'.[33]

[31] Here too, the role of intonation should not be overemphasized. In (76), the stress is on *man* and *woman*, but if we are interested in how many Dutchmen bought, rented and stole cars, then the stress in the sentence *One Dutchman bought a car and rented a car, but no Dutchman stole a car* will be on *one*, *no* and the verbs.

Partee and Rooth also consider parallel cases with *every* and *or*, e.g., (*i*) *John bought or sold every car* (transitive) and (*ii*) *Every student failed or got a D* (intransitive). In these cases, intuitions are obscured by the fact that the wide scope-*or* readings entail (i.e., are hidden by) the wide scope-*every* ones.

[32] Recall that TYPE is the category-to-type assignment of PTQ which was given in (3).

[33] On these 'widest scope-*or*' readings, a disjunction of claims is ascribed to Wanda: 'Wanda claims that John seeks a fish or Wanda claims that John seeks a bike', etc.

Such readings are beyond the reach of rule (68), but they can be attained if (68) is further generalized so as to be applicable to an arbitrary value type b of a translation of a function type $(\vec{a}, b)$, yielding a translation of type $(\vec{a}, ((s, ((s, b), c)), c))$ (cf. the operation of value raising defined below). Given this generalization of (68), we need to be able to assign ever higher types to the noun phrase *a fish* and *a bike*.

The first sentence in (77), for instance, requires an application of the generalized rule to the rightmost occurrence of t in type (69) of the translations (70) of *a fish* and *a bike*,[34] which results in noun phrase translations of type $((s, \text{TYPE(TV)}), ((s, e), ((s, \text{TYPE(PV)}), ((s, e), t))))$; the second sentence in (77) requires a second application of the rule to the translations used for the first sentence; the third sentence requires a third application; the fourth sentence ...

The strategy we will follow can be characterized as a 'reversal' of Montague's 'generalizing to the worst case'. Rather, the generalization will be to the 'best case' on the lexical level, since every lexical expression will be assigned a basic translation of the minimal type available for that particular expression. Generalized syntactic/semantic rules will permit the compounding of all translations which 'fit', and type-shifting rules will produce derived translations out of lexical and compound ones. Furthermore, the recursive nature of these rules will preclude the existence of a 'worst case'.

As we have seen, the type-shifting rules proposed by Partee and Rooth can be employed for representing various interesting scope ambiguities of coordinating expressions. It will turn out, moreover, that a generalization and extension of their rules allows one to account for coordination scope ambiguities which the original rules (including the generalized rule (68)) cannot represent,[35] as well as the quantifier scope and *de dicto/de re* ambiguities

[34] Where $b = t$, $c = ((s, e), t)$, $\vec{a} = a_1, a_2$, where $a_1 = (s, ((s, ((s, ((s, e), t)), t)), ((s, e), t)))$ and $a_2 = (s, e)$.

[35] Such as the 'widest scope-*or*' readings of the sentences in (77), and the desired *de dicto* wide scope-*or* reading of the sentence *John believes that a man or a woman won*, a problematic case for which Partee and Rooth suggest 'an alternative to the flip-flop rule [= (68)]': 'a rule quantifying in terms of a higher type' (p. 376). Notice, however, that the latter rule is *not* an alternative to the 'flip-flop' rule (68): both rules will have to be used *together*, since the term of type $((s, ((s, ((s, ((s, e), t)), t)), t)), t)$ which is to be quantified in,

$$\lambda R.[^{\vee}R(^{\wedge}\lambda P.\exists u[\text{MAN}(u) \wedge {}^{\vee}P(^{\wedge}u)]) \vee {}^{\vee}R(^{\wedge}\lambda P.\exists u[\text{WOMAN}(u) \wedge {}^{\vee}P(^{\wedge}u)])],$$

can only be obtained if (68) is allowed to apply (before generalized disjunction – and in violation of the coercion principle; a grammar which employs the two very diverse mechanisms of type-shifting rules and rules of quantification for the generation of scope

which were discussed in Section 1. So, the 'reverse' strategy can be put to use in a treatment of scope ambiguities without quantifying-in or storage.

Rules of quantification are syntactic rules which have a primarily semantic motivation. They also serve a more syntactic function (viz., the representation of anaphoric pronouns, which will be discussed in Section 6 below), but one of the main reasons for their introduction is semantic: the representation of quantifier scope and *de dicto/de re* ambiguities. From a syntactic point of view this can be seen as an exceptional interference in syntax by the semantic principle of compositionality.[36]

Another reason for looking for an alternative is the fact that the syntactic operation of these rules ('if $[_{\mathrm{T}}\alpha]$ is a syntactic variable, replace all occurrences of he_i by α; and if $[_{\mathrm{T}}\alpha]$ is not a syntactic variable, replace the first occurrence of he_i by α, and the remaining occurrences of he_i by anaphoric pronouns of appropriate case and gender') is rather complex. In several theories of syntax – including some monostratal ones which are attractive from the point of view of compositional semantics, like GPSG and its offspring[37] or flexible categorial grammar[38] –, such complex syntactic operations are disallowed.

But also for those who do not feel bound to restrictions on possible syntactic operations, there are reasons to look for alternatives. Janssen (1986, Part 2, pp. 137–172) amply dicusses the undesirable consequences of having rules of quantification and syntactic variables (the production of 'sentences' such as *John loves* he_{23} is one of the more innocent among them), and introduces a metagrammatical 'variable principle' in order to control things.[39]

---

ambiguities does not seem remarkable for attractivity, for that matter).

[36] Rules of quantification are the only rules in PTQ which give rise to syntactic derivations which do not respect the 'intuitive' syntactic constituent structure, with the result (cf. Section 1.4) that they assign counterintuitive part/whole structures.

[37] For GPSG ('generalized phrase structure grammar'), see Gazdar, Klein, Pullum and Sag (1985); Klein and Sag (1985). GPSG does not treat scope ambiguities. Gazdar, Klein, Pullum and Sag suggest using Cooper storage. HPSG ('head-driven phrase structure grammar'; cf. Pollard and Sag (1987)) has a situation-theoretic semantics.

[38] See Lambek (1958); Van Benthem (1986, Chapter 7; 1991); Moortgat (1988). The treatment of quantifier scope in this framework will be discussed in the next section.

[39] A further unattractive aspect of rules of quantification is that they tend to proliferate. The PTQ fragment has three such rules, viz., (X),i, (XI),i and (XII),i (of course, each rule comprises infinitely many instances, but that is not the point here) for quantifying in noun phrases into sentences, intransitive verb phrases, and common nouns, respectively. (The same holds for quantifier storage. In order to obtain a fragment which is equivalent to PTQ, Cooper (1975) is forced to define three corresponding NP-retrieving operations, cf.

Therefore, various attempts have been made to avoid the syntactic device of quantifying-in, while preserving the results of PTQ as regards the representation of quantifier scope ambiguities. However, Cooper storage, the main alternative in the framework of Montague grammar, has met with objections from a semantic viewpoint, since it 'takes the complication out of the syntax and puts it into the semantics' (Cooper (1983), p. 65; cf. Section 1.5 above).[40]

---

footnote 13 above.) Now, in a larger fragment one might wish to quantify in noun phrases into other categories. However, Rooth – who needs quantifying-in into noun phrases – solves this problem by defining a generalized 'cross-categorial' rule for quantifying in noun phrases into expressions of arbitrary conjoinable type, along the lines of (39) and (40) (Rooth (1985), pp. 128–138; see also Partee and Rooth (1983), pp. 380–381).

[40]Another objection against Cooper's theory involves his treatment of anaphoric pronouns. Having done away with quantifying-in and syntactic variables, Cooper treats (syntactically unindexed) pronouns as lexical NPs which are assigned the translation set $\{\langle\langle\lambda P.{}^{\vee}P(x_i)\rangle\rangle \mid i \in \mathbb{N}\}$. Hence, it cannot be seen from the pronoun itself which variable x_i has been used in a translation. A pronoun with translation $\lambda P.{}^{\vee}P(x_i)$ becomes bound when an item $\langle\delta', x_i\rangle$ is retrieved from the store, but this is syntactically invisible as well. This means that it is impossible to know which pronouns are bound by which quantifiers without interpreting them. As a consequence, the theory rests crucially 'on a non-syntactic treatment of gender agreement, since in order to know whether a sentence violates gender agreement we have to know which pronouns are bound by [which] quantifiers' (Cooper 1975, p. 207, see also pp. 66–70). This seems intuitive for languages like English which have natural gender, but it will not work for a language like German which has grammatical gender: thus, if gender agreement 'is semantics, individuals in the model [must] not only have properties like 'being male', but also like 'being denoted by a word with neutral grammatical gender'. This does not seem reasonable' (Landman and Moerdijk (1983), p. 101). Whereas Cooper (1975) stresses this point too, Cooper (1983) presents 'problems for syntactic treatments of gender', only granting that 'languages commonly have an admixture of grammatical gender and natural gender. For example, although German *Mädchen* is neuter and [grammatically requires] the pronoun *es* 'it', it is quite often possible or preferable to use the feminine pronoun *sie* 'she'.' (Cooper (1983), p. 176). Thus, (*ii*) is more acceptable than (*i*):

(*i*) Ich sah das Mädchen, als es hereinkam ('I saw the girl, when it came in')

(*ii*) Ich sah das Mädchen, als sie hereinkam ('I saw the girl, when she came in')

However, it is unclear what this is supposed to show about German gender agreement, for there are also cases where grammatical agreement is acceptable, but natural agreement is definitely out:

(*iii*) Der Lügendetektor^m sagt dem Mädchen^n daß es^n ihn^m betrügt

(*iv*) *Der Lügendetektor^m sagt dem Mädchen^n daß sie^f es^n betrügt

('The lie detector tells the girl that she cheats it'; m = masculine, f = feminine, n = neuter; example from Landman and Moerdijk (1983), pp. 100–101.) The difference between these cases is systematic, and seems to correspond to the difference between contexts which merely allow 'coference' ((*i*) and (*ii*)) and contexts which also admit 'bound anaphora' ((*iii*) and (*iv*)) (the terminology has been borrowed from Reinhart (1983)).

3 Flexibility in Syntax and Semantics

In the next section we will present a formal elaboration of the 'reverse' strategy sketched above. The traditional functional correspondence between categories and types will be given up. Instead, we will assign every expression a translation of its minimal type, and derive the translations of other types from this lexical translation by general rules, thus obtaining the flexible Montague grammar of Section 4. In the present section we will 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 directed, non-commutative version) or too strong (in its non-directed, commutative version) for the representation of scope ambiguities. The grammar of Section 4 is essentially an intensional version of a subsystem of the non-directed Lambek calculus, together with a semantics along the lines of Van Benthem.

In its original, directed version (see 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 completely determined by the categories they belong to. Restricting ourselves to the product-free part of **L**,[41] we distinguish the following categories:

(78) Let ATOM be some finite set of atomic categories.
Then CAT, the set of categories, is the smallest set such that
(*i*) ATOM $\subseteq$ CAT; and
(*ii*) if $a \in$ CAT and $b \in$ CAT, then $a/b \in$ CAT and $b\backslash a \in$ CAT.

The calculus **L** consists of a set of axioms and five inference rules. It characterizes the derivability of so-called *sequents* of categories, a notion which

The way in which rules of quantification deal with anaphoric pronouns will be discussed in Section 6 below.

[41] That is, we will not consider *product* categories $a \bullet b$ (the expressions of which consist of an expression of category a followed by an expression of category b), since these do not figure in the proposals which will be discussed below. The compound categories which do occur in (78) have the following meaning: an expression of the *right-searching functor* category a/b (pronounced 'a slash b' or 'a over b') combines with an expression of the *argument* category b on its *right*-hand side to form an expression of the *value* category a. Likewise, an expression of the *left-searching functor* category $b\backslash a$ (pronounced 'b backslash a' or 'b under a') combines with an expression of the *argument* category b on its *left*-hand side to form an expression of the *value* category a. In both cases, the argument category b constitutes the 'denominator' of the 'fraction', and the value category a makes up its 'numerator'.

is defined in (79). An expression consisting of the lexical items $e_1, \ldots, e_n$ with respective categories $c_1, \ldots, c_n$ is parsed as belonging to a certain category c if and only if the statement '$c_1, \ldots, c_n$ is a c' (written as the sequent $c_1, \ldots, c_n \vdash c$) is an axiom of **L** or can be derived as a theorem using the axioms and inference rules of **L**.

(79) A *sequent* is an expression $T \vdash c$, where T is a finite non-empty sequence of categories, and $c \in$ CAT. (So, $T = c_1, \ldots, c_n$, where $n > 0$ and for all i, $1 \leq i \leq n$: $c_i \in$ CAT.)

We will call $c_1, \ldots, c_n$ the *left-hand side*, and c the *right-hand side* or *goal* of the sequent $c_1, \ldots, c_n \vdash c$. The *axioms* and *inference rules*[42] of **L** are given in (80) and (81):

(80) AXIOM, the set of *axioms* of **L**, is the set $\{c \vdash c \mid c \in \text{CAT}\}$.

(81) $$\frac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c}\,[/L] \qquad \frac{T \vdash b \quad U, a, V \vdash c}{U, T, b\backslash a, V \vdash c}\,[\backslash L]$$

$$\frac{T, b \vdash a}{T \vdash a/b}\,[/R] \qquad \frac{b, T \vdash a}{T \vdash b\backslash a}\,[\backslash R]$$

$$\frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$$

In (81), a, b, and c denote categories and T, U, and V finite sequences of categories, of which T is non-empty.[43]

Lambek (1958) 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 $T \vdash c$, 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

[42]The rules $/L$, $\backslash L$ and $/R$, $\backslash R$ are pronounced 'slash left', 'backslash left' and 'slash right', 'backslash right', as they involve occurrences of / and \ in the left-hand side and the right-hand side of the conclusion, respectively.

[43]More formally: **L** is the smallest set of sequents such that (*i*) AXIOM $\subseteq$ **L**; (*ii*) if $T \vdash b \in$ **L** and $U, a, V \vdash c \in$ **L**, then $U, a/b, T, V \vdash c \in$ **L**; etc. Furthermore, let V be some set of lexical items, let Γ be a function assigning a finite set of categories to each member of V, and let c, the distinguished category, be some member of CAT. Then the *language recognized* by an **L** grammar $\langle V, \Gamma, c\rangle$ is the set of strings $e_1, \ldots e_n$ over V such that for all i, $1 \leq i \leq n$, there are $c_i \in \Gamma(e_i)$ such that $c_1, \ldots, c_n \vdash c \in$ **L**.

inference rules $/L$, $\backslash L$, $/R$ and $\backslash R$, the number of occurrences of / and \ in the premise(s) is strictly smaller than that in the conclusion – the conclusion contains a / or \ which is absent from the premise(s). Thus, establishing the derivability of the premise sequent(s) 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 (and interpretations) of sentences containing so-called *non-constituent coordination*, cf. (82).

(82) Mary loves and Wanda likes John

We will not dwell on this topic (see Moortgat (1988) and Barry and Pickering (1990)), but focus on the semantic interpretation of **L** instead.

That a strong link can be established between syntax and semantics in **L** has repeatedly been stressed by Van Benthem (e.g., 1986; 1988; 1991), who showed that the semantic interpretations of an **L**-derivable sequent are directly determined by the proofs of its validity in the syntactic calculus: axioms correspond to *identity*, the rules $/L$ and $\backslash L$ to *functional application*, the rules $/R$ and $\backslash R$ to *lambda abstraction*, and the *Cut* rule to *substitution*.[44]

In this semantics, every category c is associated with a type: $\text{TYPE}(c)$. For atomic categories, $\text{TYPE}(c)$ is stipulated, and $\text{TYPE}(a/b) = \text{TYPE}(b\backslash a) = (\text{TYPE}(b), \text{TYPE}(a))$.

Moreover, all categories in a sequent $c_1, \ldots, c_n \vdash c$ are assigned an interpretation. For the categories c_i $(1 \leq i \leq n)$ in the left-hand side, this interpretation is some variable v_i of type $\text{TYPE}(c_i)$; for the right-hand side category c, it is a (possibly complex) lambda term γ of type $\text{TYPE}(c)$. This assignment is usually written as $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma$.

In (83), u, v, x and w represent variables of type $\text{TYPE}(a)$, $\text{TYPE}(b)$, $\text{TYPE}(c)$ and $(\text{TYPE}(b), \text{TYPE}(a))$; α, β and γ denote terms of type $\text{TYPE}(a)$, $\text{TYPE}(b)$ and $\text{TYPE}(c)$; and T', U' and V' refer to the sequences of variables assigned to the sequences of categories T, U and V. Only the terms of the active categories are indicated separately; the terms of other categories are identical in premises and conclusion. The expression $\gamma[u{:=}\alpha]$ denotes the

[44] We will employ a modified (but equivalent) version of Moortgat's (1988) adaptation of Van Benthem's semantics to a sequent format. See Chapter 3 below for motivation.

result of substituting the term α for all free occurrences of u in γ:

(83)

SYNTAX:	SEMANTICS:
$c \vdash c$ [AXIOM]	$x \vdash x$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c}\,[/L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', w, T', V' \vdash \gamma[u{:=}w(\beta)]}$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, T, b\backslash a, V \vdash c}\,[\backslash L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', T', w, V' \vdash \gamma[u{:=}w(\beta)]}$
$\dfrac{T, b \vdash a}{T \vdash a/b}\,[/R]$	$\dfrac{T', v \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{b, T \vdash a}{T \vdash b\backslash a}\,[\backslash R]$	$\dfrac{v, T' \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$	$\dfrac{T' \vdash \alpha \quad U', u, V' \vdash \gamma}{U', T', V' \vdash \gamma[u{:=}\alpha]}$

In the context of a proof, each axiom instance is assigned a different variable v, and each instance of the rules $/L$ and $\backslash L$ introduces a fresh variable w. The interpretation which a proof of $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma$ assigns to a sequence of expressions $e_1, \ldots, e_n$ of categories $c_1, \ldots, c_n$ with interpretations $e'_1, \ldots, e'_n$ is obtained by performing the simultaneous substitution $[v_1{:=}e'_1, \ldots, v_n{:=}e'_n]$ in the term γ which is assigned to the goal of the conclusion sequent: $\gamma[v_1{:=}e'_1, \ldots, v_n{:=}e'_n]$.[45] (Examples are forthcoming.)

It can be shown that if the sequent $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma$ is derivable in **L**, then there is a *Cut*-free **L**-derivation of $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma'$ such that the terms γ and γ' are equivalent.[46] Hence we only have to consider

[45]The semantics has been kept extensional for expository purposes, but it can be intensionalized easily. Namely, (*i*) define TYPE(a/b) = TYPE($b\backslash a$) = ((s,(TYPE(b)),TYPE(a)), instead of (TYPE(b),TYPE(a)); (*ii*) assign the same terms to the axioms and the *Cut* rule; and (*iii*) change the semantic interpretations of $/L$ and $/R$ as follows: $[/L]$ if $T : T' \vdash b : \beta$ and $U : U', a : u, V : V' \vdash c : \gamma$, then $U : U', a/b : w, T : T', V : V' \vdash \gamma[u{:=}w(\hat{}\beta)]$, and $[/R]$ if $T : T', b : v \vdash a : \alpha$, then $T : T' \vdash a/b : \lambda x.\alpha[v{:=}\check{}x]$. (The rules $\backslash L$ and $\backslash R$ must be changed analogously. Here, x is a (fresh) variable of type $(s, \text{TYPE}(b))$.)

[46]The proof is straightforward: one simply shows that application of Lambek's *Cut* elimination algorithm always leads to semantically equivalent results (cf. Lambek (1990), Moortgat (1990) and Chapter 3 below).

the (finite number of) *Cut*-free derivations to get all semantic interpretations which are assigned to a sequent. But in spite of the fact that a given sequent has only finitely many *Cut*-free derivations, *Cut*-less **L** still suffers from what has been called the 'spurious ambiguity problem' (König (1989)): it supplies a proliferation of proofs for one and the same interpretation, so that in general interpretations will be computed n times instead of once. Meanwhile, the problem itself has been solved n times (cf. Hepple (1990), Leslie (1990), Moortgat (1990), Roorda (1991) and Chapter 3 below). The most conventional solution consists in normalizing the sequent calculus **L** to such an extent that always exactly one canonical representative is isolated from the class of equivalent proofs.[47] The resulting normalized calculus **L*** (*i*) only allows axiom instances $c \vdash c$ when c is an atomic category; (*ii*) prohibits major (right-hand side) premises of $\backslash L$ and $/L$ inferences which are the conclusion of a $\backslash R$ or $/R$ inference; and (*iii*) keeps the same category active when going down from axioms via major premises of $\backslash L$ and $/L$ inferences. Below we will only consider normal form **L** derivations: proofs which are isomorphic to derivations in the normalized calculus. Thus, different derivations of a sequent will always have non-equivalent terms assigned to the goal of the conclusion.

3.1 Scope Ambiguities: L

We will now consider several analyses which have been proposed for dealing with quantifier scope ambiguities in (subsystems of) **L** (see Van Benthem (1986), Dowty (1988), a.o.), and show that they are inadequate.

First, notice that the semantic results mentioned above immediately entail that under the usual analysis, according to which verbs take noun phrases – being of the atomic category np – as arguments, a sentence like (84) is unambiguous within **L**.

(84) Every man finds a woman

For if *every man* and *a woman* belong to the atomic category np, and *finds* is of category $(np\backslash s)/np$, then the sequent $np, (np\backslash s)/np, np \vdash s$ which corresponds to sentence (84) has only one *Cut*-free (and normal form) derivation,

[47] This is the method of Hepple (1990) and Chapter 3 below.

and (hence) only one interpretation:

(85)
$$\dfrac{np \vdash np \quad \dfrac{np \vdash np \quad s \vdash s}{np, np\backslash s^* \vdash s}\,[\backslash L]}{np, (np\backslash s)/np^*, np \vdash s}\,[/L]$$

The symbol '*' is used to keep track of the active category in derivations.

With different categories assigned to NPs and TVs, however, scope ambiguities can be accounted for in **L**: if the lexical categories and translations are as in (86),[48] then sentence (84) obtains the two readings in (87):

(86)

EXPRESSION	CATEGORY	TRANSLATION
every man	$s/(n\backslash s)$	$\lambda P.\forall x[\text{MAN}(x) \rightarrow P(x)]$
finds	$(n\backslash s)/n$	FIND
a woman	$(s/n)\backslash s$	$\lambda P.\exists y[\text{WOMAN}(y) \wedge P(y)]$

(87) (*i*) $\forall x[\text{MAN}(x) \rightarrow \exists y[\text{WOMAN}(y) \wedge \text{FIND}(y)(x)]]$
(*ii*) $\exists y[\text{WOMAN}(y) \wedge \forall x[\text{MAN}(x) \rightarrow \text{FIND}(y)(x)]]$

The sequent $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$ has six *Cut*-free derivations, two of which are in normal form: (88) and (89). The interpretation of the former derivation yields reading (87) (*i*), while the interpretation of the latter one results in reading (87) (*ii*).

(88)
$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad \dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}\,[\backslash L]}{n, (n\backslash s)/n^*, n \vdash s}\,[/L]}{n, (n\backslash s)/n \vdash s/n^*}\,[/R] \quad s \vdash s}{n, (n\backslash s)/n, (s/n)\backslash s^* \vdash s}\,[\backslash L]}{(n\backslash s)/n, (s/n)\backslash s \vdash n\backslash s^*}\,[\backslash R] \quad s \vdash s}{s/(n\backslash s)^*, (n\backslash s)/n, (s/n)\backslash s \vdash s}\,[/L]$$

(89)
$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad \dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}\,[\backslash L]}{n, (n\backslash s)/n^*, n \vdash s}\,[/L]}{(n\backslash s)/n, n \vdash n\backslash s^*}\,[\backslash R] \quad s \vdash s}{s/(n\backslash s)^*, (n\backslash s)/n, n \vdash s}\,[/L]}{s/(n\backslash s), (n\backslash s)/n \vdash s/n^*}\,[/R] \quad s \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^* \vdash s}\,[\backslash L]$$

[48] Throughout the present section we will assume that $\text{TYPE}(n) = e$, and $\text{TYPE}(s) = t$.

By way of illustration, we show the respective semantic interpretations of the derivations (88) and (89):

(90)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{w \vdash w \quad \dfrac{u \vdash u \quad v \vdash v}{u, x \vdash x(u)}\,[\backslash L]}{u, y, w \vdash y(w)(u)}\,[/L]}{u, y \vdash \lambda w.y(w)(u)}\,[/R] \quad z \vdash z}{u, y, w' \vdash w'(\lambda w.y(w)(u))}\,[\backslash L]}{y, w' \vdash \lambda u.w'(\lambda w.y(w)(u))}\,[\backslash R] \quad v' \vdash v'}{u', y, w' \vdash u'(\lambda u.w'(\lambda w.y(w)(u)))}\,[/L]$$

(91)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{w \vdash w \quad \dfrac{u \vdash u \quad v \vdash v}{u, x \vdash x(u)}\,[\backslash L]}{u, y, w \vdash y(w)(u)}\,[/L]}{y, w \vdash \lambda u.y(w)(u)}\,[\backslash R] \quad z \vdash z}{u', y, w \vdash u'(\lambda u.y(w)(u))}\,[/L]}{u', y \vdash \lambda w.u'(\lambda u.y(w)(u))}\,[/R] \quad v' \vdash v'}{u', y, w' \vdash w'(\lambda w.u'(\lambda u.y(w)(u)))}\,[\backslash L]$$

The results obtained by performing the simultaneous substitution (92) in the terms $u'(\lambda u.w'(\lambda w.y(w)(u)))$ and $w'(\lambda w.u'(\lambda u.y(w)(u)))$ which are assigned to the conclusion goals in (90) and (91), respectively, reduce to the readings (87) (*i*) and (*ii*).

(92) $u' := \lambda P.\forall x[\text{MAN}(x) \rightarrow P(x)]$
$y := \text{FIND}$
$w' := \lambda P.\exists y[\text{WOMAN}(y) \wedge P(y)]$

Notice, however, that the analysis given in (86) is not a very natural one, from a syntactic perspective. 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-hand side of the transitive verb, respectively. For none of the following sequents is **L**-derivable:[49]

(93) $(s/n)\backslash s, (n\backslash s)/n, s/(n\backslash s) \not\vdash s$
$(s/n)\backslash s, (n\backslash s)/n, (s/n)\backslash s \not\vdash s$
$s/(n\backslash s), (n\backslash s)/n, s/(n\backslash s) \not\vdash s$

[49] Precisely this fact prompted Lambek (1958, Section 6) to propose the category $s/(n\backslash s)$ for subject pronouns and $(s/n)\backslash s$ for direct object pronouns, thus accounting for the unacceptability of **him likes he*.

An alternative account of the scope ambiguity of (84) within **L** also assumes that noun phrases are assigned the complex categories $(s/n)\backslash s$ and $s/(n\backslash s)$, but stipulates in addition that verbs take noun phrases as arguments. 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 if $np_1 = np_2$. (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 to $s/(n\backslash s)$ and $(s/n)\backslash s$, respectively, since the sequent (94) has five normal form derivations, which are listed in (95) through (99). Each of these derivations is followed by the (reduced) terms it assigns to the categories of its conclusion sequent.

(94) $s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s \vdash s$

(95)
$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}[/R]}{s/n \vdash s/n^*}[/L] \quad s \vdash s}{s/n, (s/n)\backslash s^* \vdash s}[\backslash L]}{(s/n)\backslash s \vdash (s/n)\backslash s^*}[\backslash R] \quad \dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}[\backslash L]}{n \vdash s/(n\backslash s)^*}[/R] \quad s \vdash s}{n, (s/(n\backslash s))\backslash s^* \vdash s}[\backslash L]}{\dfrac{n, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s)^*, (s/n)\backslash s \vdash s}{((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s \vdash n\backslash s^*}[\backslash R] \quad s \vdash s}[/L]}{s/(n\backslash s)^*, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s \vdash s}[/L]$$

$x, y, z \vdash x(\lambda u.y(z)(\lambda w.w(u)))$

(96)
$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}[/L]}{n \vdash (s/n)\backslash s^*}[\backslash R] \quad \dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}[\backslash L]}{n \vdash s/(n\backslash s)^*}[/R] \quad s \vdash s}{n, (s/(n\backslash s))\backslash s^* \vdash s}[\backslash L]}{\dfrac{n, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s)^*, n \vdash s}{n, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s) \vdash s/n^*}[/R] \quad s \vdash s}[/L]}{\dfrac{n, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s^* \vdash s}{((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s \vdash n\backslash s^*}[\backslash R] \quad s \vdash s}[\backslash L]}{s/(n\backslash s)^*, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s \vdash s}[/L]$$

$x, y, z \vdash x(\lambda u.z(\lambda v.y(\lambda w.w(v))(\lambda w.w(u))))$

(97)

$$\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}\,[/L]}{s/n \vdash s/n^*}\,[/R] \qquad s \vdash s}{\dfrac{s/n, (s/n)\backslash s^* \vdash s}{(s/n)\backslash s \vdash (s/n)\backslash s^*}\,[\backslash R]}\,[\backslash L] \qquad \dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}\,[\backslash L]}{n\backslash s \vdash n\backslash s^*}\,[\backslash R] \qquad s \vdash s}{\dfrac{s/(n\backslash s)^*, n\backslash s \vdash s}{s/(n\backslash s) \vdash s/(n\backslash s)^*}\,[/R] \qquad s \vdash s}\,[/L]}{\dfrac{s/(n\backslash s), (s/(n\backslash s))\backslash s^* \vdash s}{s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s)^*, (s/n)\backslash s \vdash s}\,[/L]}\,[\backslash L]$$

$x, y, z \vdash y(z)(x)$

(98)

$$\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}\,[/L]}{n \vdash (s/n)\backslash s^*}\,[\backslash R] \qquad \dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}\,[\backslash L]}{n \vdash s/(n\backslash s)^*}\,[/R] \qquad s \vdash s}{n, (s/(n\backslash s))\backslash s^* \vdash s}\,[\backslash L]}{\dfrac{n, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s)^*, n \vdash s}{\dfrac{((s/(n\backslash s))\backslash s)/((s/n)\backslash s), n \vdash n\backslash s^* \qquad s \vdash s}{\dfrac{s/(n\backslash s)^*, ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), n \vdash s}{\dfrac{s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s) \vdash s/n^* \qquad s \vdash s}{s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s^* \vdash s}\,[\backslash L]}\,[/R]}\,[/L]}\,[\backslash R]}\,[/L]$$

$x, y, z \vdash z(\lambda v.x(\lambda u.y(\lambda w.w(v))(\lambda w.w(u))))$

(99)

$$\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}\,[/L]}{n \vdash (s/n)\backslash s^*}\,[\backslash R] \qquad \dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}\,[\backslash L]}{n\backslash s \vdash n\backslash s^*}\,[\backslash R] \qquad s \vdash s}{\dfrac{s/(n\backslash s)^*, n\backslash s \vdash s}{s/(n\backslash s) \vdash s/(n\backslash s)^*}\,[/R] \qquad s \vdash s}\,[/L]}{\dfrac{s/(n\backslash s), (s/(n\backslash s))\backslash s^* \vdash s}{\dfrac{s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s)^*, n \vdash s}{\dfrac{s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s) \vdash s/n^* \qquad s \vdash s}{s/(n\backslash s), ((s/(n\backslash s))\backslash s)/((s/n)\backslash s), (s/n)\backslash s^* \vdash s}\,[\backslash L]}\,[/R]}\,[/L]}\,[\backslash L]$$

$x, y, z \vdash z(\lambda v.y(\lambda w.w(v))(x))$

Let np_1 and np_2 abbreviate $s/(n\backslash s)$ and $(s/n)\backslash s$, respectively. When the lexical translations in (100) are substituted for x, y and z in the interpretations of the goals of the conclusions of (95) through (99), then the interpretations

of (95), (96) and (97) collapse into the wide scope-subject reading (87) (*i*), and the remaining derivations (98) and (99) represent the wide scope-object reading (87) (*ii*).[50]

(100)

EXPRESSION	CATEGORY	TRANSLATION
every man	np_1	$\lambda P.\forall x[\text{MAN}(x) \rightarrow P(x)]$
finds	$(np_1\backslash s)/np_2$	$\lambda Y\lambda X.X(\lambda x.Y(\lambda y.\text{FIND}(y)(x)))$
a woman	np_2	$\lambda P.\exists y[\text{WOMAN}(y) \wedge P(y)]$

An interesting aspect of this alternative analysis is that it predicts three readings for *Every man seeks a woman*, provided that the 'intensional' verb *seek* is assigned the UG-style translation $\lambda Y\lambda X.X(\lambda x.\text{SEEK}(Y)(x))$:[51]

(101) (*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*) $\forall x[\text{MAN}(x) \rightarrow \text{SEEK}(\lambda P.\exists y[\text{WOMAN}(y) \wedge P(y)])(x)]$

The derivations (95) and (97) lead to (*iii*), while (96) results in (*i*), and (98) and (99) arrive at (*ii*).[52]

It may seem that an adequate treatment of quantifier scope in the Lambek calculus is within reach. Still, a curious feature of the previous accounts is that the 'internal directionality' of noun phrases – $s/(n\backslash s)$ versus $(s/n)\backslash s$ – seems to be a non-negligible factor. This means that word order has an essential influence on the semantic possibilities of the calculus. When we have subject-verb-object (SVO) order as in (84), or OVS order, things turn out right.

(102) 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'

[50]The lexical translation of *finds* in (100) encodes wide scope for the subject noun phrase, but this is not essential. With lexical wide scope for the direct object noun phrase, (87) (*i*) would have been expressed by (95) and (96), and (87) (*ii*) by (97), (98) and (99).

[51]Since our semantics is extensional, this is not really an intensional translation. But cf. footnote 45.

[52]Note that the difference between (95) and (97) (or that between (98) and (99)) 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.

But in SOV cases like the embedded clause in the Dutch sentence (102) (assuming that *iedere man* and *een vrouw* have the respective categories np_1 and np_2, where $np_1, np_2 \in \{(s/n)\backslash s, s/(n\backslash s)\}$; and that *zoekt* is of category $np_2\backslash(np_1\backslash s)$), we get only two normal form derivations: (103) and (104) – that is, under 'optimal conditions': when np_1 (*iedere man*) and np_2 (*een vrouw*) are equal to $s/(n\backslash s)$ and $(s/n)\backslash s$, respectively.[53]

(103)

$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}[/L]}{s/n \vdash s/n^*}[/R] \qquad s \vdash s}{s/n, (s/n)\backslash s^* \vdash s}[\backslash L]}{(s/n)\backslash s \vdash (s/n)\backslash s^*}[\backslash R] \qquad \dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}[\backslash L]}{n\backslash s \vdash n\backslash s^*}[\backslash R] \qquad s \vdash s}{s/(n\backslash s)^*, n\backslash s \vdash s}[/L]}{s/(n\backslash s) \vdash s/(n\backslash s)^*}[/R] \qquad s \vdash s}{s/(n\backslash s), (s/(n\backslash s))\backslash s^* \vdash s}[\backslash L]$$

$$\dfrac{}{s/(n\backslash s), (s/n)\backslash s, ((s/n)\backslash s)\backslash((s/(n\backslash s))\backslash s)^* \vdash s}[\backslash L]$$

$$y, x, z \vdash z(x)(y)$$

(104)

$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{s/n^*, n \vdash s}[/L]}{s/n \vdash s/n^*}[/R] \qquad s \vdash s}{s/n, (s/n)\backslash s^* \vdash s}[\backslash L]}{(s/n)\backslash s \vdash (s/n)\backslash s^*}[\backslash R] \qquad \dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^* \vdash s}[\backslash L]}{n \vdash s/(n\backslash s)^*}[/R] \qquad s \vdash s}{n, (s/(n\backslash s))\backslash s^* \vdash s}[\backslash L]}{(s/(n\backslash s))\backslash s \vdash n\backslash s^*}[\backslash R] \qquad s \vdash s}{s/(n\backslash s)^*, (s/(n\backslash s))\backslash s \vdash s}[/L]$$

$$\dfrac{}{s/(n\backslash s), (s/n)\backslash s, ((s/n)\backslash s)\backslash((s/(n\backslash s))\backslash s)^* \vdash s}[\backslash L]$$

$$y, x, z \vdash y(\lambda v.z(x)(\lambda w.w(v)))$$

This entails that with $\lambda Y \lambda X.X(\lambda x.\textsc{seek}(Y)(x))$ as the lexical translation of *zoekt*, only (101) (*iii*) is represented.

Alternatively, one could adopt the wide scope-object encoding translation $\lambda Y \lambda X.Y(\lambda y.X(\lambda x.\textsc{seek}(\lambda P.P(y))(x)))$, but that would leave (101) (*iii*) unaccounted for. The third possibility, of assuming the lexical translation $\lambda Y \lambda X.X(\lambda x.Y(\lambda y.\textsc{seek}(\lambda P.P(y))(x)))$, would yield only (101) (*i*).

As a consequence, we are 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' (Gosse Bouma, p.c.).

[53] The same holds when both the subject and the object noun phrase are of the category $s/(n\backslash s)$. Both possibilities with subject category $(s/n)\backslash s$ give rise to just one normal form derivation (and reading).

3.2 Scope Ambiguities: LP

Since it is the directedness of **L** which thwarts us, an obvious strategy is to take a closer look at **LP** (for Lambek calculus with Permutation, also known as Lambek-Van Benthem calculus), the non-directed version of **L** which has been studied extensively by Van Benthem (cf., e.g., his (1991)). A possible formalization of **LP** and its semantics is shown in (105). In **LP**, the two kinds of compound categories in **L**, the left- and right-searching functors a/b and $b\backslash a$, are collapsed into one category (b, a), a non-directed functor from argument categories b to value categories a. (Note the direct correspondence between syntactic categories and semantic types.) **LP** preserves the axioms and the *Cut* rule of **L**; it contains rules for the introduction of the connective $(,)$ in the left-hand side (L) and the right-hand side (R) of sequents; and a permutation rule P is added, which allows one to interchange two adjacent categories. (Every permutation is a composition of interchanges between neighbours.)

(105)

SYNTAX:	SEMANTICS:
$c \vdash c$ [AXIOM]	$x \vdash x$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, (b, a), T, V \vdash c}\,[L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', w, T', V' \vdash \gamma[u := w(\beta)]}$
$\dfrac{T, b \vdash a}{T \vdash (b, a)}\,[R]$	$\dfrac{T', v \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{U, a, b, V \vdash c}{U, b, a, V \vdash c}\,[P]$	$\dfrac{U', u, v, V' \vdash \gamma}{U', v, u, V' \vdash \gamma}$
$\dfrac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$	$\dfrac{T' \vdash \alpha \quad U', u, V' \vdash \gamma}{U', T', V' \vdash \gamma[u := \alpha]}$

Conceived of as a syntactic theory, i.e., as a theory of syntactic flexibility or *category* change, **LP** is not really interesting, because it has the following property: if a sequent $T \vdash c$ is **LP**-derivable, then for every permutation $\pi(T)$ of the sequence T: $\pi(T) \vdash c$ is **LP**-derivable as well.

Nevertheless, **LP** might be useful as a semantic theory, i.e., as a theory of semantic flexibility or *type* change. We might, for example, restrict ourselves to sequents of categories which are derivable in our favourite syntactic

theory (**L**, say), and let **LP** take care of their semantic interpretation. More formally:

(106) The compound expression $e_1, \ldots, e_n$ is of category c and translates as $\gamma[v_1 := \gamma_1, \ldots, v_n := \gamma_n]$ of type τ iff for all i, $1 \leq i \leq n$:
(i) e_i is of category c_i and translates as γ_i of type τ_i;
(ii) $c_1, \ldots, c_n \vdash_{\mathbf{L}} c$; and
(iii) $\tau_1 : v_1, \ldots, \tau_n : v_n \vdash_{\mathbf{LP}} \tau : \gamma$.

But now the permuting effects of **LP** will turn up in the semantics. Compare (107) and (108). As we have seen, the sequent $np, (np\backslash s)/np, np \vdash s$ is **L**-derivable. In **LP** there are two non-equivalent derivations of the 'semantic sequent' $e, (e, (e, t)), e \vdash t$: (109) and (110) – which is one too many.

(107) John loves Paris

(108)

EXPRESSION	CATEGORY	TYPE	TRANSLATION
John	np	e	j
loves	$(np\backslash s)/np$	$(e,(e,t))$	LOVE
Paris	np	e	p

(109)
$$\dfrac{\dfrac{\dfrac{\dfrac{e \vdash e \quad \dfrac{t \vdash t}{(e,t)^*, e \vdash t}\,[L]}{e \vdash e \quad (e,t)^*, e \vdash t}\,[L]}{(e,(e,t))^*, e, e \vdash t}\,[P]}{(e,(e,t)), e^*, e^* \vdash t}\,[P]}{e^*, (e,(e,t))^*, e \vdash t}\,[P]$$

$x, y, z \vdash y(z)(x)$ (LOVE$(p)(j)$: 'John loves Paris')

(110)
$$\dfrac{\dfrac{\dfrac{\dfrac{e \vdash e \quad \dfrac{t \vdash t}{(e,t)^*, e \vdash t}\,[L]}{e \vdash e \quad (e,t)^*, e \vdash t}\,[L]}{(e,(e,t))^*, e, e \vdash t}\,[P]}{e^*, (e,(e,t))^*, e \vdash t}\,[P]$$

$x, y, z \vdash y(x)(z)$ (LOVE$(j)(p)$: 'Paris loves John')

Moreover, given the assignments in (112), sentence (111) gets two readings of type t, cf. (113) and (114).

(111) Every man walks

(112)

EXPRESSION	CATEGORY	TYPE	TRANSLATION
every	np/cn	$((e,t),((e,t),t))$	$\lambda Q \lambda P.\forall x[Q(x) \rightarrow P(x)]$
man	cn	(e,t)	MAN
walks	$np\backslash s$	(e,t)	WALK

(113)

$$\dfrac{(e,t) \vdash (e,t) \quad \dfrac{(e,t) \vdash (e,t) \quad t \vdash t}{((e,t),t)^*, (e,t) \vdash t}\,[L]}{((e,t),((e,t),t))^*, (e,t), (e,t) \vdash t}\,[L]$$

$$x, y, z \vdash x(y)(z)$$

(114)

$$\dfrac{\dfrac{(e,t) \vdash (e,t) \quad \dfrac{(e,t) \vdash (e,t) \quad t \vdash t}{((e,t),t)^*, (e,t) \vdash t}\,[L]}{((e,t),((e,t),t))^*, (e,t), (e,t) \vdash t}\,[L]}{((e,t),((e,t),t)), (e,t)^*, (e,t)^* \vdash t}\,[P]$$

$$x, y, z \vdash x(z)(y)$$

While derivation (113) yields the interpretation $\forall x[\text{MAN}(x) \rightarrow \text{WALK}(x)]$, derivation (114) supplies $\forall x[\text{WALK}(x) \rightarrow \text{MAN}(x)]$, which expresses that *only* men walk.

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 'θ-roles' of the verb and the order of the arguments of the determiner *every*. (In this particular case, the symmetry of the determiner *a* avoids another multiplication of the number of readings by two.)

4 The Fragment

In the present section we will develop a calculus of semantic flexibility which is not thwarted by the directedness of syntactic categories (since it is a theory of semantic – i.e., non-directed – *types*), but at the same time avoids the pervasive semantic permutation of the calculus **LP**.[54] We will not couch

[54]We have, of course, not shown that it is altogether impossible to treat quantifier scope ambiguities within flexible categorial grammar; only that – contrary to what has been suggested – current (sub)systems (of **L** or **LP**) are inadequate. Thus, Moortgat (1988) simply incorporates the system of Hendriks (1988) into **L**; Moortgat (1993) presents a more principled sequent approach (see also Morrill (1993)); and Emms (1990, 1992) offers an interesting alternative using polymorphic quantifiers and variable categories.

this semantic theory in a sequent format, but attach it to the Montagovian fragment defined in Section 1.

First, every *syntactic category* C is not associated with a single type, but with an infinite set of types: the *type set* $T(C)$ of category C. This type set consists of a basic type, $t_B(C)$, together with types which are derived from the basic type by general rules. For a flexible version of the fragment introduced in Section 1, we choose the basic types specified in (115):

(115)

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

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

(116)
- (*i*) $t_B(C) \in T(C)$;
- (*ii*) if $(\vec{a},b) \in T(C)$, then $(\vec{a},((s,((s,b),d)),d)) \in T(C)$;
- (*iii*) if $(\vec{a},((s,b),(\vec{c},d))) \in T(C)$, then $(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d))) \in T(C)$; and
- (*iv*) if $(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d))) \in T(C)$, then $(\vec{a},((s,b),(\vec{c},d))) \in T(C)$.

The type s in definition (116) is the specific **IL** type associated with the set of possible worlds, whereas b and d represent arbitrary types. Moreover, the symbols $\vec{a}$ and $\vec{c}$ represent arbitrary sequences $a_1,\ldots,a_n$ and $c_1,\ldots,c_m$ of types such that $n \geq 0$ and $m \geq 0$; and if b is a type, then $(\vec{a},b)$ and $(\vec{c},b)$ represent the types $(a_1,(\ldots(a_n,b)\ldots))$ and $(c_1,(\ldots(c_m,b)\ldots))$, respectively.

The clauses (*ii*), (*iii*) and (*iv*) in (116) mirror the operations of value raising, argument raising and argument lowering which will be defined in (126) through (128) below.

Some examples: if clause (*ii*) is applied to $t_B(\mathrm{T}) = e$ (with $\vec{a}$ instantiated as the empty sequence; b as e; and d as t), then the type $((s,((s,e),t)),t)$ – TYPE(T) in PTQ – turns out to be a derived type in the type set $T(\mathrm{T})$ of T, the category of noun phrases.

Further, if clause (*iii*) is applied to $t_B(\text{TV}) = ((s,e),((s,e),t))$ (with $\vec{a}$ instantiated as the empty sequence; b as e; $\vec{c}$ as the sequence consisting of the single type (s,e); and d as t), then the type $((s,((s,((s,e),t)),t)),((s,e),t))$ – PTQ's TYPE(TV) – appears as a member of $T(\text{TV})$.

Finally, notice that there are two different ways of showing that the type $((s,((s,((s,e),t)),t)),t)$ – the type of IVs in UG (cf. Section 2 above) – is a member of $T(\text{IV})$: one can apply clause (*ii*) (with $\vec{a}$ instantiated as the empty sequence; b as $((s,e),t)$; and d as t) or clause (*iii*) (with both $\vec{a}$ and $\vec{c}$ instantiated as the empty sequence; b as e; and d as t) to the type $t_B(\text{IV}) = ((s,e),t)$.

Second, every – lexical or compound – *expression* α of category C is not assigned one translation, but a *translation set* $Tr(\alpha)$: a set consisting of one or more basic translations plus derived translations.

4.1 Basic Translations

Every *lexical* expression α is assigned a set $Tr_B(\alpha)$ of basic translations in the lexicon. Typically – that is: for all lexically unambiguous expressions – $Tr_B(\alpha)$ will only contain one member, $tr_B(\alpha)$, so that $Tr_B(\alpha) = \{tr_B(\alpha)\}$. The non-basic members of $Tr(\alpha)$ are derived from $Tr_B(\alpha)$ by general rules which will be defined in Section 4.2 below. It is important to stress that $tr_B(\alpha)$ *is not necessarily of the basic type* $t_B(C)$ *associated with the category* C *to which* α *belongs.*

The fragment contains the lexical basic translations listed in (117) (where the determiner type $((s,((s,e),t)),((s,((s,e),t)),t))$ is abbreviated as δ):

(117)

α	$tr_B(\alpha)_{\text{type of } tr_B(\alpha)}$
[CN *price*]	$\text{PRICE}_{((s,e),t)}$
[CN *man*]	$\lambda x.[\text{MAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]_{((s,e),t)}$
[CN *woman*]	$\lambda x.[\text{WOMAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]_{((s,e),t)}$
[CN *unicorn*]	$\lambda x.[\text{UNICORN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]_{((s,e),t)}$
[IV *is missing*]	$\text{BE-MISSING}_{((s,((s,((s,e),t)),t)),t)}$
[IV *walks*]	$\lambda x.\text{WALK}(^{\vee}x)_{((s,e),t)}$
[IV *rises*]	$\text{RISE}_{((s,e),t)}$
[T *John*]	j_e

$[_{\mathrm{T}}\mathit{Mary}]$	m_e
$[_{\mathrm{Det}}\mathit{a}]$	$\lambda P\lambda Q.\exists x[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)]_{\delta}$
$[_{\mathrm{Det}}\mathit{the}]$	$\lambda P\lambda Q.\exists x[\forall y[{}^{\vee}P(y) \leftrightarrow x = y] \wedge {}^{\vee}Q(x)]_{\delta}$
$[_{\mathrm{Det}}\mathit{every}]$	$\lambda P\lambda Q.\forall x[{}^{\vee}P(x) \rightarrow {}^{\vee}Q(x)]_{\delta}$
$[_{\mathrm{Det}}\mathit{one}]$	$\lambda P\lambda Q.\exists x\forall y[[{}^{\vee}P(y) \wedge {}^{\vee}Q(y)] \leftrightarrow x = y]_{\delta}$
$[_{\mathrm{TV}}\mathit{seeks}]$	$\lambda T\lambda x.\mathrm{SEEK}(T)({}^{\vee}x)_{((s,((s,(s,e),t)),t)),((s,e),t))}$
$[_{\mathrm{TV}}\mathit{finds}]$	$\lambda y\lambda x.\mathrm{FIND}({}^{\vee}y)({}^{\vee}x)_{((s,e),((s,e),t))}$
$[_{\mathrm{PV}}\mathit{claims\ that}]$	$\lambda p\lambda x.\mathrm{CLAIM}(p)({}^{\vee}x)_{((s,t),((s,e),t))}$
$[_{\mathrm{PV}}\mathit{believes\ that}]$	$\lambda p\lambda x.\mathrm{BELIEVE}(p)({}^{\vee}x)_{((s,t),((s,e),t))}$

Observe (*i*) that the fragment does not contain syntactic variables he_i; (*ii*) that the basic translation of *seek* is not of type $((s,e),((s,e),t))$, the basic type $t_B(\mathrm{TV})$ of transitive verb phrases; (*iii*) that the basic translation of *be missing* is not of type $((s,e),t)$, the basic type $t_B(\mathrm{IV})$ of intransitive verb phrases; and (*iv*) 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 type e, the basic type $t_B(\mathrm{T})$ of noun phrases, but of the derived noun phrase type $((s,((s,e),t)),t)$.

Compound expressions γ are also assigned a set $Tr_B(\gamma)$ of basic translations. This set is determined by:

1. the respective translation sets $Tr(\alpha_1),\ldots,Tr(\alpha_n)$ consisting of both the basic translations and the derived translations of the immediate constituent expressions $\alpha_1,\ldots,\alpha_n$ of γ; and
2. the semantic operation R associated with the syntactic rule which constructs γ from its immediate constituent expressions $\alpha_1,\ldots,\alpha_n$.

Since the fragment only contains binary rules, all compound expressions γ in the fragment are built up from exactly two immediate constituent expressions: α and β. Hence the above can be summarized as follows:

(118) $Tr_B(\gamma) = \{R(\alpha',\beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$

There are three (binary) *semantic operations* R in the fragment:[55]

(119) FA ((Intensional) Functional Application):
for α' of type $((s,a),b)$ and β' of type a: $\mathrm{FA}(\alpha',\beta') = \alpha'(\hat{}\beta')$

[55] Recall from (39) and (40) above that for $\vec{a} = a_1,\ldots,a_n$:

- $(\vec{a},b)$ denotes the type $(a_1,(\ldots(a_n,b)\ldots))$;
- $\vec{x}_{\vec{a}}$ denotes $v_{1,a_1},\ldots,v_{n,a_n}$ (for all $1 \leq i \leq n$: v_i is the i-th variable of type a_i);

(120) GC (Generalized Conjunction):
for α' and β' of type $(\vec{a}, t)$: $\mathrm{GC}(\alpha', \beta') = \lambda\vec{x}_{\vec{a}}.[\alpha'(\vec{x}) \wedge \beta'(\vec{x})]$

(121) GD (Generalized Disjunction):
for α' and β' of type $(\vec{a}, t)$: $\mathrm{GD}(\alpha', \beta') = \lambda\vec{x}_{\vec{a}}.[\alpha'(\vec{x}) \vee \beta'(\vec{x})]$

Note that the generalized operations FA, GC and GD are *partial* in that they do not yield a value for all translations α' and β'. For $\mathrm{FA}(\alpha', \beta')$ to be defined, there has to be a type a such that α' is of type $((s, a), b)$ and β' is of type a, while $\mathrm{GC}(\alpha', \beta')$ and $\mathrm{GD}(\alpha', \beta')$ are only defined if α' and β' are of the same conjoinable type $(\vec{a}, t)$. The operations defined in (119), (120) and (121) figure in the *rules of the fragment*:

RULES OF APPLICATION
$Tr_B(\gamma) = \{\mathrm{FA}(\alpha', \beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$
(I) $[_{\mathrm{T}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{S}}[_{\mathrm{T}}\alpha][_{\mathrm{IV}}\beta]]$
(II) $[_{\mathrm{TV}}\alpha] + [_{\mathrm{T}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{IV}}[_{\mathrm{TV}}\alpha][_{\mathrm{T}}\beta]]$
(III) $[_{\mathrm{Det}}\alpha] + [_{\mathrm{CN}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{T}}[_{\mathrm{Det}}\alpha][_{\mathrm{CN}}\beta]]$
(IV) $[_{\mathrm{PV}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{IV}}[_{\mathrm{PV}}\alpha][_{\mathrm{S}}\beta]]$

RULES OF CONJUNCTION
$Tr_B(\gamma) = \{\mathrm{GC}(\alpha', \beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$
(V) $[_{\mathrm{S}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{S}}[_{\mathrm{S}}\alpha]$ *and* $[_{\mathrm{S}}\beta]]$
(VI) $[_{\mathrm{IV}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{IV}}[_{\mathrm{IV}}\alpha]$ *and* $[_{\mathrm{IV}}\beta]]$
(VII) $[_{\mathrm{TV}}\alpha] + [_{\mathrm{TV}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{TV}}[_{\mathrm{TV}}\alpha]$ *and* $[_{\mathrm{TV}}\beta]]$
(VIII) $[_{\mathrm{CN}}\alpha] + [_{\mathrm{CN}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{CN}}[_{\mathrm{CN}}\alpha]$ *and* $[_{\mathrm{CN}}\beta]]$

RULES OF DISJUNCTION
$Tr_B(\gamma) = \{\mathrm{GD}(\alpha', \beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$
(IX) $[_{\mathrm{S}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{S}}[_{\mathrm{S}}\alpha]$ *or* $[_{\mathrm{S}}\beta]]$
(X) $[_{\mathrm{IV}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{IV}}[_{\mathrm{IV}}\alpha]$ *or* $[_{\mathrm{IV}}\beta]]$
(XI) $[_{\mathrm{T}}\alpha] + [_{\mathrm{T}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{T}}[_{\mathrm{T}}\alpha]$ *or* $[_{\mathrm{T}}\beta]]$
(XII) $[_{\mathrm{TV}}\alpha] + [_{\mathrm{TV}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{TV}}[_{\mathrm{TV}}\alpha]$ *or* $[_{\mathrm{TV}}\beta]]$
(XIII) $[_{\mathrm{CN}}\alpha] + [_{\mathrm{CN}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{CN}}[_{\mathrm{CN}}\alpha]$ *or* $[_{\mathrm{CN}}\beta]]$

There are no rules of quantification in the fragment. Instead, the representation of quantifier scope ambiguities will be taken care of by the derived

- $\lambda\vec{x}_{\vec{a}}.\tau$ denotes $\lambda v_{1,a_1} \ldots \lambda v_{n,a_n}.\tau$; and
- $\tau(\vec{x}_{\vec{a}})$ denotes $\tau(v_{1,a_1}) \ldots (v_{n,a_n})$.

translations (see Section 4.2 below). Further, the rules (VII), (VIII), (XII) and (XIII) have been added, which permit the construction of conjunctions and disjunctions of transitive verb phrases and common noun phrases.

Consider, by way of example, the noun phrase *a fish*. The rule of application (III) allows us to construct this compound noun phrase from the determiner *a* and the common noun *fish*. On the semantic side (see (117) above), we have that:

(122) $tr_B(a) = \lambda P \lambda Q.\exists y[^\vee P(y) \wedge {}^\vee Q(y)]$ and
$tr_B(\textit{fish}) = \lambda x.[\textsc{fish}(^\vee x) \wedge \exists u(x = {}^\wedge u)]$

Moreover, rule (III) specifies that:

(123) $Tr_B(a\ \textit{fish}) = \{\, \mathrm{FA}(\alpha', \beta') \mid \alpha' \in Tr(a) \text{ and } \beta' \in Tr(\textit{fish}) \,\}$

Because $tr_B(a) \in Tr(a)$ and $tr_B(\textit{fish}) \in Tr(\textit{fish})$, we know that the term $\mathrm{FA}(tr_B(a), tr_B(\textit{fish}))$ of type $((s,((s,e),t)),t)$ – which is written out in (124) – is a member of $Tr_B(a\ \textit{fish})$.

(124) $[\lambda P \lambda Q.\exists y[^\vee P(y) \wedge {}^\vee Q(y)]](^\wedge \lambda x.[\textsc{fish}(^\vee x) \wedge \exists u(x = {}^\wedge u)])$

And (124) is equivalent to $\lambda Q.\exists v[\textsc{fish}(v) \wedge {}^\vee Q(^\wedge v)]$.

4.2 Derived Translations

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

(125) $Tr_B(\alpha) \subseteq Tr(\alpha)$;

(126) VR (Value Raising)
if $\alpha' \in Tr(\alpha)$ and α' is of type $(\vec{a}, b)$, then
$\lambda \vec{x}_{\vec{a}} \lambda u_{(s,((s,b),d))}.{}^\vee u(^\wedge[\alpha'(\vec{x})]) \in Tr(\alpha)$;

(127) AR (Argument Raising)
if $\alpha' \in Tr(\alpha)$ and α' is of type $(\vec{a}, ((s,b),(\vec{c},d)))$, then
$\lambda \vec{x}_{\vec{a}} \lambda w_{(s,((s,((s,b),d)),d))} \lambda \vec{y}_{\vec{c}}.{}^\vee w(^\wedge \lambda z_{(s,b)}.\alpha'(\vec{x})(z)(\vec{y})) \in Tr(\alpha)$; and

(128) AL (Argument Lowering)
if $\alpha' \in Tr(\alpha)$ and α' is of type $(\vec{a}, ((s, ((s,((s,b),d)),d)),(\vec{c},d)))$,
then $\lambda \vec{x}_{\vec{a}} \lambda z_{(s,b)} \lambda \vec{y}_{\vec{c}}.\alpha'(\vec{x})(^\wedge \lambda u_{(s,((s,b),d))}.{}^\vee u(z))(\vec{y}) \in Tr(\alpha)$.

Recall that $\vec{a}$ and $\vec{c}$ are sequences $a_1, \ldots, a_n$ and $c_1, \ldots, c_m$ of types such that $n \geq 0$ and $m \geq 0$, and that if b is a type, then $(\vec{a}, b)$ and $(\vec{c}, b)$ represent the types $(a_1, (\ldots (a_n, b) \ldots))$ and $(c_1, (\ldots (c_m, b) \ldots))$.

In addition, the symbol $\vec{x}_{\vec{a}}$ denotes the sequence of variables $x_1, \ldots, x_n$ where for $1 \leq i \leq n$: x_i is the i-th variable of type a_i (so that $\vec{x}_{\vec{a}} = v_{1,a_1}, \ldots, v_{n,a_n}$); the symbol $\vec{y}_{\vec{c}}$ denotes the sequence of variables $y_1, \ldots, y_m$ where for $1 \leq i \leq m$: y_i is the $(n+1+i)$-th variable of type a_i (so that $\vec{y}_{\vec{c}} = v_{n+2,c_1}, \ldots, v_{n+1+m,c_m}$); and z, u and w denote the variables $v_{n+1,(s,b)}$, $v_{n+1,(s,((s,b),d))}$ and $v_{n+1,(s,((s,((s,b),d)),d))}$, respectively.

Finally, if τ is a term, then the expressions $\lambda\vec{x}_{\vec{a}}.\tau$, $\lambda\vec{y}_{\vec{c}}.\tau$, $\tau(\vec{x}_{\vec{a}})$ and $\tau(\vec{y}_{\vec{c}})$ abbreviate $\lambda x_1 \ldots \lambda x_n.\tau$, $\lambda y_1 \ldots \lambda y_m.\tau$, $\tau(x_1) \ldots (x_n)$ and $\tau(y_1) \ldots (y_m)$, respectively.

Observe that – reflecting definition (116) – the compound terms given in (126), (127) and (128) are of the respective types $(\vec{a}, ((s, ((s, b), d)), d))$, $(\vec{a}, ((s, ((s, ((s, b), d)), d)), (\vec{c}, d)))$ and $(\vec{a}, ((s, b), (\vec{c}, d)))$; and note that all variables displayed in these terms are different. For instance, the term assigned by value raising is the following one:

$$\lambda v_{1,a_1} \ldots \lambda v_{n,a_n} \lambda v_{n+1,(s,((s,b),d))}.{}^{\vee}v_{n+1,(s,((s,b),d))}({}^{\wedge}[\alpha'(v_{1,a_1}) \ldots (v_{n,a_n})])$$

Some lexical examples:

If value raising is applied to the basic translation j of type e of the noun phrase *John* – with $\vec{a}$ instantiated as the empty sequence; b as e; and d as t –, we get the PTQ translation $\lambda P.{}^{\vee}P({}^{\wedge}j)$ of type $((s, ((s, e), t)), t)$ as a derived translation.

If argument raising is applied to (the first argument of) the basic translation $\lambda y \lambda x.\text{FIND}({}^{\vee}y)({}^{\vee}x)$ of type $((s, e), ((s, e), t))$ of *finds* – with $\vec{a}$ instantiated as the empty sequence; b as e; $\vec{c}$ as the sequence consisting of the single type (s, e); and d as t –, the PTQ translation $\lambda T \lambda x.{}^{\vee}T({}^{\wedge}\lambda y.\text{FIND}({}^{\vee}y)({}^{\vee}x))$ of type $((s, ((s, ((s, e), t)), t)), ((s, e), t))$ is obtained.[56]

If argument lowering is applied to (the first argument of) the basic translation $\lambda T \lambda x.\text{SEEK}(T)({}^{\vee}x)$ of type $((s, ((s, ((s, e), t)), t)), ((s, e), t))$ of *seeks* – with $\vec{a}$ instantiated as the empty sequence; b as e; d as t; and $\vec{c}$ as the sequence consisting of the single type (s, e) –, then a derived translation results which can be applied to the intension of the basic translations of proper

[56]Strictly speaking, the translation obtained is not the PTQ translation, but the equivalent expression $\lambda T \lambda x.{}^{\vee}T({}^{\wedge}\lambda y.[\lambda y \lambda x.\text{FIND}({}^{\vee}y)({}^{\vee}x)](y)(x))$, which can be reduced to the PTQ translation via two λ-conversions. In the sequel we will continue to speak roughly about reduced (basic) translations as if they were (basic) translations themselves.

names (which are of type e); this derived translation of type $((s,e),((s,e),t))$ reduces to $\lambda y \lambda x.\text{SEEK}(\hat{}\lambda P.\check{}P(y))(\check{}x)$.

As was noted above, the semantic operations FA, GC and GD are *partial* with respect to $Tr(\alpha)$ and $Tr(\beta)$. Sometimes, $\text{FA}(\alpha',\beta')$, $\text{GC}(\alpha',\beta')$ and $\text{GD}(\alpha',\beta')$ fail to be defined for $\alpha' \in Tr(\alpha)$ and $\beta' \in Tr(\beta)$. For instance, $Tr_B(\textit{John and Mary})$ does not contain $\text{GC}(j,m)$, simply because j and m are of type e, which is not a conjoinable type $(\vec{a},t)$. Nor does the basic translation set $Tr_B(\textit{seeks John})$ contain the result of applying FA to $\lambda T \lambda x.\text{SEEK}(T)(\check{}x)$, an expression of type $((s,((s,((s,e),t)),t)),((s,e),t))$, and j of type e, the reason being that $\text{FA}(\lambda T \lambda x.\text{SEEK}(T)(\check{}x),j)$ is undefined: $\lambda T \lambda x.\text{SEEK}(T)(\check{}x)$ 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 $\lambda P.\check{}P(\hat{}j)$ and $\lambda P.\check{}P(\hat{}m)$ of type $((s,((s,e),t)),t)$ at our disposal, the basic translation sets $Tr_B(\textit{John and Mary})$ and $Tr_B(\textit{seeks John})$ contain at least the following members:[57]

(129) $\text{GC}(\lambda P.\check{}P(\hat{}j),\lambda P.\check{}P(\hat{}m)) =$
$\lambda Q.[\lambda P.\check{}P(\hat{}j)(Q) \wedge \lambda P.\check{}P(\hat{}m))(Q)]$

(130) $\text{FA}(\lambda T \lambda x.\text{SEEK}(T)(\check{}x),\lambda P.\check{}P(\hat{}j)) =$
$[\lambda T \lambda x.\text{SEEK}(T)(\check{}x)](\hat{}\lambda P.\check{}P(\hat{}j))$

The following compound example shows that the fragment accounts for the *de dicto* wide scope-*or* reading (67) of sentence (62), *John seeks a fish or a bike*, in that it yields a translation which reduces to (67):

Rule (III) enables us to construct *a fish* from the determiner *a* and the common noun *fish*. In the previous section we have seen that the term $\text{FA}(tr_B(a),tr_B(\textit{fish}))$ of type $((s,((s,e),t)),t)$, which is repeated in (131) and which is equivalent to $\lambda Q.\exists v[\text{FISH}(v) \wedge \check{}Q(\hat{}v)]$, is a member of the set $Tr_B(\textit{a fish})$:

(131) $[\lambda P \lambda Q.\exists y[\check{}P(y) \wedge \check{}Q(y)]](\hat{}\lambda x.[\text{FISH}(\check{}x) \wedge \exists u(x = \hat{}u)])$

[57] That every (lexical or compound) expression defined by the fragment has a non-empty translation set is easily established, e.g., by checking that every expression of category S, T, CN, IV, Det, TV and PV has at least one translation of the (conjoinable) type t, e^2, e^1, $((s,e^2),t)$, $((s,e^1),e^2)$, $((s,e^2),((s,e^2),t))$ and $((s,t),((s,e^2),t))$, respectively. (The e^n-notation is defined in (132).) See the Appendix for more general results concerning the combinability of translations.

Hence, according to (125), it is also a member of $Tr(a\ fish)$. Let us assume the following notation convention for types:

(132) For a a type and $n \in \mathbb{N}$: $a^0 = a$;
$a^{n+1} = ((s, a^n), t)$.

Note, for instance, that on account of (132), the type $((s,((s,e),t)),t)$ of (131) can be written as e^2; and that $e^4 = ((s,e^3),t) = ((s,((s,e^2),t)),t) = ((s,((s,((s,e^1),t)),t)),t) = ((s,((s,((s,((s,e),t)),t)),t)),t)$.

Now, due to clause (126) (value raising), we know that $Tr(a\ fish)$ furthermore contains the derived translation (133) of type $((s,((s,e^2),t)),t)$, i.e., type e^4:

(133) $\lambda V_{(s,e^3)}.{}^{\vee}V({}^{\wedge}\mathrm{FA}(tr_B(a), tr_B(\mathit{fish})))$

Translation (133) is equivalent to (134):

(134) $\lambda V_{(s,e^3)}.{}^{\vee}V({}^{\wedge}\lambda P.\exists v[\textsc{fish}(v) \wedge {}^{\vee}P({}^{\wedge}v)])$

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

(135) $\lambda V_{(s,e^3)}.{}^{\vee}V({}^{\wedge}\lambda P.\exists v[\textsc{bike}(v) \wedge {}^{\vee}P({}^{\wedge}v)])$

The rule of disjunction (XI) specifies GD((134),(135)), which is written out in (136), as one of the basic translations[58] of the conjoined noun phrase *a fish or a bike*:

(136) $\lambda V_{(s,e^3)}.[[\lambda V.{}^{\vee}V({}^{\wedge}\lambda P.\exists v[\textsc{fish}(v) \wedge {}^{\vee}P({}^{\wedge}v)])](V)$
$\vee\, [\lambda V.{}^{\vee}V({}^{\wedge}\lambda P.\exists v[\textsc{bike}(v) \wedge {}^{\vee}P({}^{\wedge}v)])](V)]$

Translation (136) reduces to (137):

(137) $\lambda V_{(s,e^3)}.[{}^{\vee}V({}^{\wedge}\lambda P.\exists v[\textsc{fish}(v) \wedge {}^{\vee}P({}^{\wedge}v)])$
$\vee\, {}^{\vee}V({}^{\wedge}\lambda P.\exists v[\textsc{bike}(v) \wedge {}^{\vee}P({}^{\wedge}v)])]$

The terms in (136) and (137) are also of type e^4. Next, we can use argument raising (127) (applying it to the first argument of the verb) for blowing up the basic translation $\lambda T\lambda x.\textsc{seek}(T)({}^{\vee}x)$ of *seeks* which is of type $((s,e^2),((s,e),t))$. Thus we obtain (138) of type $((s,e^4),((s,e),t))$:

(138) $\lambda W_{(s,e^4)}\lambda x_{(s,e)}.{}^{\vee}W({}^{\wedge}\lambda T_{(s,e^2)}.[\lambda T\lambda x.\textsc{seek}(T)({}^{\vee}x)](T)(x))$

[58] Recall footnote 56.

Translation (138) reduces to (139):

(139) $\lambda W_{(s,e^4)}\lambda x_{(s,e)}.{}^{\vee}W(^{\wedge}\lambda T.\text{SEEK}(T)({}^{\vee}x))$

Because (137) and (139) are (equivalent to) members of the respective translation sets *Tr*(*a fish or a bike*) and *Tr*(*seeks*), we can apply rule (II), which specifies that FA((139),(137)), a term of type $((s,e),t)$ given in (140), is one of the basic translations of *seeks a fish or a bike*:

(140) $[\lambda W_{(s,e^4)}\lambda x_{(s,e)}.{}^{\vee}W(^{\wedge}\lambda T.\text{SEEK}(\text{T})({}^{\vee}x))]$
$(^{\wedge}\lambda V_{(s,e^3)}.[{}^{\vee}V(^{\wedge}\lambda P.\exists v[\text{FISH}(v) \wedge {}^{\vee}P(^{\wedge}v)])$
$\vee\, {}^{\vee}V(^{\wedge}\lambda P.\exists v[\text{BIKE}(v) \wedge {}^{\vee}P(^{\wedge}v)])])$

Translation (140) is equivalent to (141):

(141) $\lambda x_{(s,e)}.[\text{SEEK}(^{\wedge}\lambda P.\exists v[\text{FISH}(v) \wedge {}^{\vee}P(^{\wedge}v)])({}^{\vee}x)$
$\vee\, \text{SEEK}(^{\wedge}\lambda P.\exists v[\text{BIKE}(v) \wedge {}^{\vee}P(^{\wedge}v)])({}^{\vee}x)]$

Finally, we can apply rule (I), which guarantees that FA((141), j) of type t is one of the basic translations of *John seeks a fish or a bike*:

(142) $[\lambda x_{(s,e)}.[\text{SEEK}(^{\wedge}\lambda P.\exists v[\text{FISH}(v) \wedge {}^{\vee}P(^{\wedge}v)])({}^{\vee}x)$
$\vee\, \text{SEEK}(^{\wedge}\lambda P.\exists v[\text{BIKE}(v) \wedge {}^{\vee}P(^{\wedge}v)])({}^{\vee}x)]](^{\wedge}j)$

Translation (142) reduces to (67), the *de dicto* wide scope *or*-reading of *John seeks a fish or a bike* which is repeated here:

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

5 Applications

In this section we will present a number of applications of the flexible Montague grammar defined in Section 4. First, we will show in Section 5.1 that ambiguities due to the occurrence of quantified noun phrases are represented, not only in 'flat' structures (where all quantified noun phrases are immediate arguments of one relation – cf. Section 5.1.1), but also in embedded application structures of arbitrary complexity (cf. Section 5.1.2). Next, in Section 5.2, we will pay attention to ambiguities which involve the scope of coordination. Finally, it will be shown in Section 5.3 that structural restrictions on possible scopings can be accounted for within the flexible framework.

In the sequel we will focus on the availability of readings for sentences (a reading being an interpretation expressed by a translation of type t). Notice, however, that on account of the flexible set-up of the fragment, each expression is assigned translations of infinitely many types, and that, moreover, in general an interpretation is expressed by an infinite number of different but equivalent translations.

The sentence *John walks*, for example, has translations of type t^{2n} for all natural numbers n:

(144) 0. $[\lambda x.\text{WALK}(^{\vee}x)](^{\wedge}j)$
1. $\lambda V_{(s,t^1)}.{}^{\vee}V(^{\wedge}[[\lambda x.\text{WALK}(^{\vee}x)](^{\wedge}j)])$
2. $\lambda W_{(s,t^3)}.{}^{\vee}W(^{\wedge}[\lambda V.{}^{\vee}V(^{\wedge}[[\lambda x.\text{WALK}(^{\vee}x)](^{\wedge}j)])])$
3. ...

For each $n \in \mathbb{N}$, translation (144) n. of type t^{2n} is the result of n applications of value raising (with $\vec{a}$ empty and $d = t$) to FA($tr_B(walks)$,$tr_B(John)$).

Furthermore, even the translations of type $t^0 = t$ constitute an infinite set, which (properly) includes the items of the following list:

(145) 0. $[\lambda x.\text{WALK}(^{\vee}x)](^{\wedge}j)$
1. $[\lambda I_{(s,e^2)}.{}^{\vee}I(^{\wedge}\lambda x.\text{WALK}(^{\vee}x))](^{\wedge}\lambda J_{(s,e^1)}.{}^{\vee}J(^{\wedge}j))$
2. $[\lambda I'_{(s,e^4)}.{}^{\vee}I'(^{\wedge}\lambda I.{}^{\vee}I(^{\wedge}\lambda x.\text{WALK}(^{\vee}x)))](^{\wedge}\lambda J'_{(s,e^3)}.{}^{\vee}J'(^{\wedge}\lambda J.{}^{\vee}J(^{\wedge}j)))$
3. ...

For each $n \in \mathbb{N}$, translation (145) n. of type t is the result of applying FA to the terms resulting from n applications of value raising (with $\vec{a}$ empty and $d = t$) to $tr_B(walks)$ and $tr_B(John)$, respectively.

Nonetheless, all translations in (145) – as well as all other translations of type t which are assigned to *John walks* – can be reduced to WALK(j), so that the sentence has only one reading: all translations of type t assigned to the sentence express the same interpretation.

In order to know which set of readings is assigned to a sentence, or, more generally, which set of interpretations of a certain type is assigned to an expression, it is therefore necessary to determine which translations of that type assigned to that expression express the same interpretation. This task is performed in the Appendix, which supplies (the necessary and sufficient ingredients for) the proofs of claims about (numbers of) sentence readings made in the present section.

5.1 The Scope of Quantification

5.1.1 Flat Structures

Consider sentence (146), which consists of an extensional transitive verb and two proper names:

(146) John finds Mary

If the rule of application (II) is used to combine the basic translations of *finds* and *Mary*, then *finds Mary* is assigned the translation in (147), which reduces to (148):

(147) $[\lambda y \lambda x.\text{FIND}(^{\vee}y)(^{\vee}x)](^{\wedge}m)$

(148) $\lambda x.\text{FIND}(m)(^{\vee}x)$

And if, using rule (I), the term in (148) is applied to the intension of the basic translation j of *John*, then the resulting sentence (146) is translated as (149), which is equivalent to $\text{FIND}(m)(j)$.

(149) $[\lambda x.\text{FIND}(m)(^{\vee}x)](^{\wedge}j)$

It can be shown that this is the only interpretation of type t which is assigned to sentence (146) by the flexible grammar. That is: if, instead of using the basic translations, we were to combine arbitrary derived translations into a term of type t, then the result would be equivalent. So, sentence (146) has one reading.

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

(150) John seeks Mary

We will sketch two possible ways to obtain this reading.

One can apply the basic translation of *seeks*, $\lambda T \lambda x.\text{SEEK}(T)(^{\vee}x)$, to the intension of the derived translation $\lambda P.^{\vee}P(^{\wedge}m)$ of *Mary* (the result of value-raising m to type e^2) and apply the result to the intension of the basic translation j of *John*:

(151) $[[\lambda T \lambda x.\text{SEEK}(T)(^{\vee}x)](^{\wedge}\lambda P.^{\vee}P(^{\wedge}m))](^{\wedge}j)$

Alternatively, one can obtain (152) (which reduces to (153)) by applying argument lowering to the first argument of $\lambda T\lambda x.\text{SEEK}(T)(^{\vee}x)$:

(152) $\lambda y\lambda x.[\lambda T\lambda x.\text{SEEK}(T)(^{\vee}x)](^{\wedge}\lambda P.^{\vee}P(y))(x)$

(153) $\lambda y\lambda x.\text{SEEK}(^{\wedge}\lambda P.^{\vee}P(y))(^{\vee}x)$

Next, the term in (153) can be applied to the intensions of the basic translations of the proper names *John* and *Mary*, respectively:

(154) $[[\lambda y\lambda x.\text{SEEK}(^{\wedge}\lambda P.^{\vee}P(y))(^{\vee}x)](^{\wedge}m)](^{\wedge}j)$

Both (151) and (154) reduce to (155):

(155) $\text{SEEK}(^{\wedge}\lambda P.^{\vee}P(^{\wedge}m))(j)$

If the noun phrases surrounding the transitive verbs are quantified noun phrases, then the grammar assigns a different number of readings to the sentences with *seeks* and *finds*:

(156) Every man finds a unicorn

(157) (*i*) $\forall v[\text{MAN}(v) \rightarrow \exists u[\text{UNICORN}(u) \wedge \text{FIND}(u)(v)]]$
(*ii*) $\exists u[\text{UNICORN}(u) \wedge \forall v[\text{MAN}(v) \rightarrow \text{FIND}(u)(v)]]$

(158) Every man seeks a unicorn

(159) (*i*) $\forall v[\text{MAN}(v) \rightarrow \exists u[\text{UNICORN}(u) \wedge \text{SEEK}(^{\wedge}\lambda P.^{\vee}P(^{\wedge}u))(v)]]$
(*ii*) $\exists u[\text{UNICORN}(u) \wedge \forall v[\text{MAN}(v) \rightarrow \text{SEEK}(^{\wedge}\lambda P.^{\vee}P(^{\wedge}u))(v)]]$
(*iii*) $\forall v[\text{MAN}(v) \rightarrow \text{SEEK}(^{\wedge}\lambda P.\exists u[\text{UNICORN}(u) \wedge {}^{\vee}P(^{\wedge}u)])(v)]$

Sentence (156) gets the two readings expressed in (157), whereas sentence (158) can be understood in the three ways listed in (159).

As for (156): the basic translation $\lambda y\lambda x.\text{FIND}(^{\vee}y)(^{\vee}x)$ of *finds* is of type $((s,e),((s,e),t))$, and there are two non-equivalent ways, (160) and (161), to argument-raise it in two steps to the type $((s,e^2),((s,e^2),t))$:

(160) $((s,e),((s,e),t)) \Rightarrow ((s,e^2),((s,e),t)) \Rightarrow ((s,e^2),((s,e^2),t))$

(161) $((s,e),((s,e),t)) \Rightarrow ((s,e),((s,e^2),t)) \Rightarrow ((s,e^2),((s,e^2),t))$

Let α abbreviate $\lambda y \lambda x.\text{FIND}(^{\vee}y)(^{\vee}x)$. Then (160) – raising the first and the second argument in that order – yields (162), while the reverse order (161) results in (163). The terms (162) and (163) reduce to (164) and (165), respectively.

(162) $\lambda T_1 \lambda T_2.{}^{\vee}T_2(^{\wedge}\lambda x_2.[\lambda T_1 \lambda x_2.{}^{\vee}T_1(^{\wedge}\lambda x_1.[\alpha](x_1)(x_2))](T_1)(x_2))$

(163) $\lambda T_1 \lambda T_2.{}^{\vee}T_1(^{\wedge}\lambda x_1.[\lambda x_1 \lambda T_2.{}^{\vee}T_2(^{\wedge}\lambda x_2.[\alpha](x_1)(x_2))](x_1)(T_2))$

(164) $\lambda T_1 \lambda T_2.{}^{\vee}T_2(^{\wedge}\lambda x_2.{}^{\vee}T_1(^{\wedge}\lambda x_1.\text{FIND}(^{\vee}x_1)(^{\vee}x_2)))$

(165) $\lambda T_1 \lambda T_2.{}^{\vee}T_1(^{\wedge}\lambda x_1.{}^{\vee}T_2(^{\wedge}\lambda x_2.\text{FIND}(^{\vee}x_1)(^{\vee}x_2)))$

Application of (164) to the intensions of $\lambda Q.\exists u[^{\vee}\text{UNICORN}(u) \wedge {}^{\vee}Q(^{\wedge}u)]$ and $\lambda Q.\forall v[^{\vee}\text{MAN}(v) \rightarrow {}^{\vee}Q(^{\wedge}v)]$ – (reduced versions of) the basic translations of the noun phrases *every man* and *a unicorn* which result from applying the basic translation of the determiners to the intension of the basic translation of the common nouns – yields a term which reduces to (157) (*i*); and (157) (*ii*) is equivalent to the result of combining (165) with these basic translations. Again, additional type change does not lead to additional readings.

The combination of an n-ary relation of type $((s,e)_1,(\ldots((s,e)_n,t)))$ with n 'simple' quantified noun phrases[59] has maximally $n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$ readings of type t. (The number $n!$ is a maximum because noun phrases of type $((s,((s,e),t)),t)$ which result from value-raising noun phrases of type e (proper names), for instance, do not multiply the number of readings.)

In the case of sentence (158), we can apply argument lowering to the basic translation $\lambda T \lambda x.\text{SEEK}(T)(^{\vee}x)$ of *seek*. The result reduces to:

(166) $\lambda y \lambda x.\text{SEEK}(^{\wedge}\lambda P.{}^{\vee}P(y))(^{\vee}x)$

If (166) undergoes the type transitions indicated in (160) and (161), then the results (162) and (163) – this time with α representing (166) – reduce to (167) and (168), which, combined with the noun phrase interpretations, yield the readings (159) (*i*) and (159) (*ii*), respectively:

(167) $\lambda T_1 \lambda T_2.{}^{\vee}T_2(^{\wedge}\lambda x_2.{}^{\vee}T_1(^{\wedge}\lambda x_1.\text{SEEK}(^{\wedge}\lambda P.{}^{\vee}P(x_1))(^{\vee}x_2)))$

[59]That is to say: noun phrases of type $((s,((s,e),t)),t)$ which do not contain conjunctions, disjunctions or relative clauses.

(168) $\lambda T_1 \lambda T_2.{}^{\vee}T_1({}^{\wedge}\lambda x_1.{}^{\vee}T_2({}^{\wedge}\lambda x_2.\text{SEEK}({}^{\wedge}\lambda P.{}^{\vee}P(x_1))({}^{\vee}x_2)))$

But there is also a third possibility here. Starting with the basic translation $\lambda T \lambda x.\text{SEEK}(T)({}^{\vee}x)$ of *seek*, we can raise its second argument and obtain a term of type $((s,e^2),((s,e^2),t))$ which reduces to the following one:

(169) $\lambda T_1 \lambda T_2.{}^{\vee}T_2({}^{\wedge}\lambda x_2.\text{SEEK}(T_1)({}^{\vee}x_2))$

This derived translation can be used for the *de dicto* reading (159) (*iii*).

Another application of argument lowering involves sentence (170), which is assigned the two readings in (171):

(170) A man is missing

(171) (*i*) $\text{BE-MISSING}({}^{\wedge}\lambda P.\exists u[\text{MAN}(u) \wedge {}^{\vee}P({}^{\wedge}u)])$
(*ii*) $\exists u[\text{MAN}(u) \wedge \text{BE-MISSING}({}^{\wedge}\lambda P.{}^{\vee}P({}^{\wedge}u))]$

The *de dicto* reading (171) (*i*) can be obtained by combining the basic translations of the lexical expressions. The *de re* reading (171) (*ii*) results when the basic translation BE-MISSING of the verb is subjected to argument lowering and argument raising in that order:

(172) $((s,e^2),t) \Rightarrow ((s,e),t) \Rightarrow ((s,e^2),t)$

The outcome of this type shift is equivalent to (173), which, when applied to the intension of $\lambda P.\exists u[\text{MAN}(u) \wedge {}^{\vee}P({}^{\wedge}u))]$, yields a term equivalent to (171) (*ii*):

(173) $\lambda T.{}^{\vee}T({}^{\wedge}\lambda x.\text{BE-MISSING}({}^{\wedge}\lambda P.{}^{\vee}P(x)))$

On the other hand, the basic translation and the derived translation (173) of *is missing* lead to equivalent terms when they are applied to the value-raised translation of a proper name such as $\lambda P.{}^{\vee}P({}^{\wedge}j)$: the sentence *John is missing* is not ambiguous.

5.1.2 Embedded Structures

Using the flexible grammar of Section 4, it is possible to represent quantifier scope ambiguities in arbitrarily complex structures of functional application. Consider, for example, the syntactic structure of the compound sentence *Fred claims that every schoolboy believes that a mathematician wrote 'Through the Looking Glass'*:

(174) $[_S[_T\text{Fred}][_{IV}[_{PV}\text{claims that}][_S[_T[_{Det}\text{every}][_{CN}\text{schoolboy}]]$
$[_{IV}[_{PV}\text{believes that}][_S[_T[_{Det}\text{a}][_{CN}\text{mathematician}]][_{IV}[_{TV}\text{wrote}]$
$[_T\text{Through the looking glass}]]]]]]]$

With the basic translations listed below, the structure in (174) 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* with respect to the compound intensional context (175) on the other:

Fred	f_e
claims that	$\lambda p\lambda x.\text{CLAIM}(^\vee x, p)_{((s,t),((s,e),t))}$
every schoolboy	$\lambda P.\forall v[\text{BOY}(v) \rightarrow {}^\vee P(^\wedge v)]_{((s,((s,e),t)),t)}$
believes that	$\lambda p\lambda x.\text{BELIEVE}(^\vee x, p)_{((s,t),((s,e),t))}$
a mathematician	$\lambda P.\exists u[\text{MATH}(u) \wedge {}^\vee P(^\wedge u)]_{((s,((s,e),t)),t)}$
wrote	$\lambda y\lambda x.\text{WRITE}(^\vee x, {}^\vee y)_{((s,e),((s,e),t))}$
Through the looking glass	L_e

(175) [... *claims that* [... *believes that* [...]]]

(176) $\text{CLAIM}(f, {}^\wedge\forall v[\text{BOY}(v) \rightarrow \text{BELIEVE}(v, {}^\wedge\exists u[\text{MATH}(u) \wedge \text{WRITE}(u, L)])])$

(177) $\text{CLAIM}(f, {}^\wedge\forall v[\text{BOY}(v) \rightarrow \exists u[\text{MATH}(u) \wedge \text{BELIEVE}(v, {}^\wedge\text{WRITE}(u, L))]])$

(178) $\text{CLAIM}(f, {}^\wedge\exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \text{BELIEVE}(v, {}^\wedge\text{WRITE}(u, L))]])$

(179) $\exists u[\text{MATH}(u) \wedge \text{CLAIM}(f, {}^\wedge\forall v[\text{BOY}(v) \rightarrow \text{BELIEVE}(v, {}^\wedge\text{WRITE}(u, L))])]$

(180) $\forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, {}^\wedge\text{BELIEVE}(v, {}^\wedge\exists u[\text{MATH}(u) \wedge \text{WRITE}(u, L)]))]$

(181) $\forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, {}^\wedge\exists u[\text{MATH}(u) \wedge \text{BELIEVE}(v, {}^\wedge\text{WRITE}(u, L))])]$

(182) $\forall v[\text{BOY}(v) \rightarrow \exists u[\text{MATH}(u) \wedge \text{CLAIM}(f, {}^\wedge\text{BELIEVE}(v, {}^\wedge\text{WRITE}(u, L)))]]$

(183) $\exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, \hat{}\text{BELIEVE}(v, \hat{}\text{WRITE}(u, L)))]]$

By way of illustration, we show the derivation of (183), semantically the most dislocated reading of sentence (174).[60]

[60]The remaining seven readings can be obtained by applying the following type shifts:

(176)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t),((s,e^2),t))$
(177)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,e),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t^2))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e^2),t))$
(178)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,e),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t^2))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t),((s,e^2),t))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e^2),t))$
(179)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,e),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t^2))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t),((s,e^2),t))$
			$\overset{\text{VR}}{\Longrightarrow}$	$((s,t),((s,e^2),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e^2),t^2))$
	claims that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t))$
(180)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,t),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t),((s,e^2),t^2))$
	claims that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t))$
(181)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,e),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t^2))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t))$
			$\overset{\text{VR}}{\Longrightarrow}$	$((s,t^2),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e^2),t^2))$
	claims that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t))$
(182)	*wrote* :	$((s,e),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,e),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,e),((s,e^2),t^2))$
	believes that :	$((s,t),((s,e),t))$	$\overset{\text{VR}}{\Longrightarrow}$	$((s,t),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t^2))$
			$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e^2),t^2))$
	claims that :	$((s,t),((s,e),t))$	$\overset{\text{AR}}{\Longrightarrow}$	$((s,t^2),((s,e),t))$,

The noun phrase *a mathematician* must 'escape' from its original scope domain, the one embedded in *believes that*. This can be effectuated by applying value raising (creating the 'escape hatch' – i.e., w in (185)) and argument raising (letting the subject argument 'go out' – viz., via T in (186)) in that order to the basic translation of *wrote*, which is the main functor in that domain. The type shift involved is given in (184):

(184) $((s,e),((s,e),t)) \Rightarrow ((s,e),((s,e),t^2)) \Rightarrow ((s,e),((s,e^2),t^2))$

Note that $t^2 = ((s,t^1),t) = ((s,((s,t),t)),t)$. The application of value raising to the basic translation of *wrote* leads to a term which is equivalent to the following one:

(185) $\lambda y_{(s,e)} \lambda x_{(s,e)} \lambda w_{(s,t^1)} .{}^{\vee} w({}^{\wedge}\text{WRITE}({}^{\vee}x, {}^{\vee}y))$

And the result of the applying argument raising to (185) reduces to:

(186) $\lambda y_{(s,e)} \lambda T_{(s,e^2)} \lambda w_{(s,t^1)} .{}^{\vee} T({}^{\wedge}\lambda x .{}^{\vee} w({}^{\wedge}\text{WRITE}({}^{\vee}x, {}^{\vee}y)))$

The noun phrase *every schoolboy* is to have wider scope than *claims that*. The main functor of the domain embedded in *claims that* is *believes that*. We can perform the same trick here: first apply value raising, and then raise the second argument of *believes that*. Besides, due to our application of value raising to the basic translation of *wrote*, the compound first argument of *believes that* (the sentence *a mathematician wrote 'Through the Looking Glass'*) does not 'fit' anymore: being of type t^2 instead of t, it has become too 'large'. This calls for an additional application of argument raising to *believes that*, this time involving the first argument. Since *a mathematician* is going to have wider scope than *every schoolboy*, this application must *follow* both other type shifts, cf. (187):

(187) $((s,t),((s,e),t)) \Rightarrow ((s,t),((s,e),t^2)) \Rightarrow ((s,t),((s,e^2),t^2))$
$\Rightarrow ((s,t^2),((s,e^2),t^2))$

Applying the respective type shifts in (187) to the basic translation of *believes that* leads to terms which reduce to (188), (189) and (190):

(188) $\lambda p_{(s,t)} \lambda x_{(s,e)} \lambda w_{(s,t^1)} .{}^{\vee} w({}^{\wedge}\text{BELIEVE}({}^{\vee}x, p))$

---

and combining the resulting translations by means of intensional functional application.

(189) $\lambda p_{(s,t)} \lambda T_{(s,e^2)} \lambda w_{(s,t^1)}.{}^{\vee}T(\hat{}\lambda x.{}^{\vee}w(\hat{}\text{BELIEVE}({}^{\vee}x, p)))$

(190) $\lambda V_{(s,t^2)} \lambda T_{(s,e^2)} \lambda w_{(s,t^1)}.{}^{\vee}V(\hat{}\lambda p.{}^{\vee}T(\hat{}\lambda x.{}^{\vee}w(\hat{}\text{BELIEVE}({}^{\vee}x, p))))$

As a consequence of our applying value raising to *believes that*, also the compound first argument of *claims that* (the sentence *every schoolboy believes that a mathematician wrote 'Through the Looking Glass'*) has become too 'large' for its embedding verb. This necessitates a raising of the first argument of *claims that* which results in the following derived translation of type $((s, t^2), ((s, e), t))$:

(191) $\lambda V_{(s,t^2)} \lambda x_{(s,e)}.{}^{\vee}V(\hat{}\lambda p.\text{CLAIM}({}^{\vee}x, p))$

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

(192) $[\lambda y \lambda T \lambda w.{}^{\vee}T(\hat{}\lambda x.{}^{\vee}w(\hat{}\text{WRITE}({}^{\vee}x, {}^{\vee}y)))](\hat{}L)$
$(\hat{}\lambda P.\exists u[\text{MATH}(u) \wedge {}^{\vee}P(\hat{}u)])$

Translation (192) reduces to (193):

(193) $\lambda w.\exists u[\text{MATH}(u) \wedge {}^{\vee}w(\hat{}\text{WRITE}(u, L))]$

Moreover, *every schoolboy believes that a mathematician wrote 'Through the Looking Glass'* is translated as:

(194) $[\lambda V \lambda T \lambda w.{}^{\vee}V(\hat{}\lambda p.{}^{\vee}T(\hat{}\lambda x.{}^{\vee}w(\hat{}\text{BELIEVE}({}^{\vee}x, p))))](\hat{}(193))$
$(\hat{}\lambda P.\forall v[\text{BOY}(v) \rightarrow {}^{\vee}P(\hat{}v)])$

Translation (194) is equivalent to (195):

(195) $\lambda w.\exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow {}^{\vee}w(\hat{}\text{BELIEVE}(v, \hat{}\text{WRITE}(u, L)))]]$

Finally, we get the following translation for sentence (174) as a whole:

(196) $[\lambda Q \lambda x.{}^{\vee}Q(\hat{}\lambda p.\text{CLAIM}({}^{\vee}x, p))](\hat{}(195))(\hat{}f)$

And translation (196) reduces to (183), which is repeated here as (197):

(197) $\exists u[\text{MATH}(u) \wedge \forall v[\text{BOY}(v) \rightarrow \text{CLAIM}(f, \hat{}\text{BELIEVE}(v, \hat{}\text{WRITE}(u, L)))]]$

5.2 The Scope of Coordination

It was shown in Section 4.2 that the *de dicto* wide scope-*or* reading (199) (*i*) of example (198) can be represented:

(198) John seeks a fish or a bike

In sum, this sentence is assigned three readings:

(199) (*i*) $\text{SEEK}(\hat{}\lambda P.\exists v[\text{FISH}(v) \wedge \check{}P(\hat{}v)])(j)$
$\vee\ \text{SEEK}(\hat{}\lambda P.\exists v[\text{BIKE}(v) \wedge \check{}P(\hat{}v)])(j)$

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

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

It is obvious that the 'standard' *de dicto* reading (*ii*) is accounted for too. For constructing this reading, it suffices to combine the basic translations of the lexical items. The generalized disjunction of $\text{FA}(tr_B(a), tr_B(\textit{fish}))$ and $\text{FA}(tr_B(a), tr_B(\textit{bike}))$ reduces to (200):

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

And the result of applying the basic translation $\lambda T\lambda x.\text{SEEK}(T)(\check{}x))$ of *seeks* to the intension of (200) and the intension of the basic translation j of *John*, respectively, is a term which can be reduced to (199) (*ii*).

The *de re* reading (199) (*iii*) is constructed in the same way, except that instead of the basic translation of *seeks* the following derived translation has to be used:

(201) $\lambda T\lambda x.\check{}T(\hat{}\lambda y.\text{SEEK}(\hat{}\lambda P.\check{}P(y))(\check{}x))$

Translation (201) is a translation of *seeks* which results from lowering and raising the first argument of the basic translation in that order, cf. the type shift in (202):

(202) $((s, e^2), ((s, e), t)) \Rightarrow ((s, e), ((s, e), t)) \Rightarrow ((s, e^2), ((s, e), t))$

Now, consider the following minimal variant of sentence (198):

(203) John seeks a fish or bike

This example is assigned exactly the same readings as sentence (198). Again, the *de dicto* reading (199) (*ii*) is obtained by combining the basic translations of the lexical expressions, while the *de re* reading (199) (*iii*) requires the derived translation of *seeks* in (201).

In order to get at the *de dicto* wide scope-*or* reading (199) (*i*), one has to apply value raising to *fish* and *bike*, shifting the type of these common nouns from e^1 to e^3 and obtaining the following derived translations:

(204) $\lambda T_{(s,e^2)}.{}^{\vee}T({}^{\wedge}\lambda x.[\text{FISH}({}^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)])$ and
$\lambda T_{(s,e^2)}.{}^{\vee}T({}^{\wedge}\lambda x.[\text{BIKE}({}^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)])$

For **IL** constants α of type (e,t), the term $\lambda x.[\alpha({}^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$ will be abbreviated as α^* in the sequel.

Rule (XIII) specifies that the generalized disjunction of the translations in (204) is a translation of the compound common noun *fish or bike*. This translation reduces to:

(205) $\lambda T_{(s,e^2)}.[{}^{\vee}T({}^{\wedge}\text{FISH}^*) \vee {}^{\vee}T({}^{\wedge}\text{BIKE}^*)]$

Next, we apply value raising and argument raising to the determiner *a*. These raisings shift the type of this expression along the lines of (206) and yield a derived translation which is equivalent to (207):

(206) $((s,e^1),e^2)) \Rightarrow ((s,e^1),e^4) \Rightarrow ((s,e^3),e^4)$

(207) $\lambda W_{(s,e^3)}\lambda V_{(s,e^3)}.{}^{\vee}W({}^{\wedge}\lambda P.{}^{\vee}V({}^{\wedge}\lambda Q.\exists x[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)]))$

The result of applying (207) to the intension of (205) is a term of type e^4 which turns out to be equivalent to (208):

(208) $\lambda V_{(s,e^3)}.[{}^{\vee}V({}^{\wedge}\lambda Q.\exists v[\text{FISH}(v) \wedge {}^{\vee}Q({}^{\wedge}v)])$
$\vee {}^{\vee}V({}^{\wedge}\lambda Q.\exists v[\text{BIKE}(v) \wedge {}^{\vee}Q({}^{\wedge}v)])]$

Note that translation (208) of *a fish or bike* is identical to (137), the translation of *a fish or a bike* we arrived at in Section 4.2. Consequently, the *de dicto* wide scope-*or* reading (199) (*i*) can be obtained by proceeding in the same way as we did there: raise the first argument of the basic translation of *seeks* and apply the resulting derived translation to the intension of (208) and the intension of j in that order.

A related scope ambiguity is exhibited by the following sentence, which has the two readings in (210):[61]

(209) Every man and woman walks

(210) (*i*) $\forall v[[\text{MAN}(v) \wedge \text{WOMAN}(v)] \rightarrow \text{WALK}(v)]$
(*ii*) $\forall v[\text{MAN}(v) \rightarrow \text{WALK}(v)] \wedge \forall v[\text{WOMAN}(v) \rightarrow \text{WALK}(v)]$

Reading (210) (*i*), which expresses that every hermaphrodite walks, requires a minimal amount of type shifts: only the argument of *walks* needs to be raised. The more natural reading (210) (*ii*) – the availability of which led Van Eijck (1991) to an analysis in terms of a syntactically discontinuous determiner *every ... and* of type $((s,e^1),((s,e^1),e^2))$ – can be obtained in a way analogous to the wide scope-*or* reading (199) (*i*) of sentence (203): apply value raising and generalized conjunction to the basic translations of the common nouns *man* and *woman*, and raise the first argument of the determiner *every* – see (211) and (212), respectively:

(211) $\lambda T_{(s,e^2)}.[{}^{\vee}T({}^{\wedge}\text{MAN}^*) \wedge {}^{\vee}T({}^{\wedge}\text{WOMAN}^*)]$

(212) $\lambda W_{(s,e^3)}\lambda Q_{(s,e^1)}.{}^{\vee}W({}^{\wedge}\lambda P.\forall x[{}^{\vee}P(x) \rightarrow {}^{\vee}Q(x)]))$

Application of (212) to the intension of (211) provides an interpretation of the subject noun phrase *every man and woman* which reduces to (213):

(213) $\lambda Q_{(s,e^1)}.[\forall v[\text{MAN}(v) \rightarrow {}^{\vee}Q({}^{\wedge}v)] \wedge \forall v[\text{WOMAN}(v) \rightarrow {}^{\vee}Q({}^{\wedge}v)]]$

Reading (210) (*ii*) is arrived at if the basic translation of *walks* is argument-raised to type e^3 and applied to the intension of (210).

Returning to the cases of IV and TV conjunction discussed by Partee and Rooth,[62] note that in the absence of the coercion principle sentence (214) is indeed assigned two readings:

(214) John caught and ate a fish

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

[61] Example (203) was brought to my attention by Peter Ruhrberg. Sentence (209) is structurally identical to example (106) in Van Eijck (1991) and to example (35) in Partee and Rooth (1983), who attribute a variant of the present treatment of reading (210) (*ii*) to Robin Cooper.

[62] See (44), (51), (53), (55), (57) and (58) above.

The extensional transitive verbs *caught* and *ate* have basic translations of type $((s,e),((s,e),t))$, and translations of type $((s,e^2),((s,e),t))$ can be derived from these basic translations by raising their first argument:

(216) $\lambda T_{(s,e^2)}\lambda x_{(s,e)}.{}^{\vee}T({}^{\wedge}\lambda y.\textsc{catch}({}^{\vee}y)({}^{\vee}x))$ and
$\lambda T_{(s,e^2)}\lambda x_{(s,e)}.{}^{\vee}T({}^{\wedge}\lambda y.\textsc{eat}({}^{\vee}y)({}^{\vee}x))$

Reading (215) (*i*), which is not accepted by Partee and Rooth, employs the generalized conjunction of these derived translations:

(217) $\lambda T_{(s,e^2)}\lambda x_{(s,e)}.[{}^{\vee}T({}^{\wedge}\lambda y.\textsc{catch}({}^{\vee}y)({}^{\vee}x)) \wedge {}^{\vee}T({}^{\wedge}\lambda y.\textsc{eat}({}^{\vee}y)({}^{\vee}x))]$

Reading (215) (*ii*) can be achieved by first taking the generalized conjunction of the basic translations, the result of which reduces to (218), and then raising the first argument of (218), which leads to a translation equivalent to (219):

(218) $\lambda y_{(s,e)}\lambda x_{(s,e)}.[\textsc{catch}({}^{\vee}y)({}^{\vee}x) \wedge \textsc{eat}({}^{\vee}y)({}^{\vee}x)]$

(219) $\lambda T_{(s,e^2)}\lambda x_{(s,e)}.{}^{\vee}T({}^{\wedge}\lambda y.[\textsc{catch}({}^{\vee}y)({}^{\vee}x) \wedge \textsc{eat}({}^{\vee}y)({}^{\vee}x)])$

Apart from their 'primary' *de dicto* readings, the sentences with intensional verbs are also assigned *de re* readings. Sentence (220), for example, gets the three readings in (221):

(220) John needed and bought a coat

(221) (*i*) $\textsc{need}({}^{\wedge}\lambda P.\exists u[\textsc{coat}(u) \wedge {}^{\vee}P({}^{\wedge}u)])(j)$
$\wedge \exists u[\textsc{coat}(u) \wedge \textsc{buy}(u)(j)]$
(*ii*) $\exists u[\textsc{coat}(u) \wedge \textsc{need}({}^{\wedge}\lambda P.{}^{\vee}P({}^{\wedge}u))(j)]$
$\wedge \exists u[\textsc{coat}(u) \wedge \textsc{buy}(u)(j)]$
(*iii*) $\exists u[\textsc{coat}(u) \wedge [\textsc{need}({}^{\wedge}\lambda P.{}^{\vee}P({}^{\wedge}u))(j) \wedge \textsc{buy}(u)(j)]]$

Suppose that the transitive verbs have the respective basic translations $\lambda T\lambda x.\textsc{need}(T)({}^{\vee}x)$ of type $((s,e^2),((s,e),t))$ and $\lambda y\lambda x.\textsc{buy}({}^{\vee}y)({}^{\vee}x)$ of type $((s,e),((s,e),t))$. Then reading (221) (*i*) can be obtained by argument-raising the basic translation of *bought* to type $((s,e^2),((s,e),t))$ and applying generalized conjunction at that level. Reading (221) (*ii*) requires an application of argument lowering to the basic translation of *needed* followed by a raising of both verbs to type $((s,e^2),((s,e),t))$ before they are conjoined. Reading (221) (*iii*) comes out if the basic translation of *needed* is

argument-lowered and generalized conjunction takes place at the level of type $((s,e),((s,e),t))$.

Moreover, observe that the desired *de dicto* wide scope-*or* reading (223) of sentence (222), the problematic case for which Partee and Rooth suggest 'a rule quantifying in terms of a higher type' (cf. footnote 35 above), is accounted for:

(222) John believes that a man or a woman walks

(223) $\text{BELIEVE}(j, {}^\wedge\exists u[\text{MAN}(u) \wedge \text{WALK}(u)])$
$\vee\ \text{BELIEVE}(j, {}^\wedge\exists u[\text{WOMAN}(u) \wedge \text{WALK}(u)])$

For the generation of reading (223), the noun phrases *a man* and *a woman* have to be value-raised to type e^4. The generalized disjunction of the resulting derived translations reduces to (224).

(224) $\lambda V_{(s,e^3)}.[{}^\vee V({}^\wedge\lambda P.\exists u[\text{MAN}(u) \wedge {}^\vee P({}^\wedge u)])$
$\vee\ {}^\vee V({}^\wedge\lambda P.\exists u[\text{WOMAN}(u) \wedge {}^\vee P({}^\wedge u)])]$

Further, applying argument raising, value raising and argument raising in that order (cf. the type shift in (225)) to the basic translation $\lambda x.\text{WALK}({}^\vee x)$ of *walks* produces the interpretation in (226):

(225) $((s,e),t) \Rightarrow ((s,e^2),t) \Rightarrow ((s,e^2),t^2) \Rightarrow ((s,e^4),t^2)$

(226) $\lambda W_{(s,e^4)}\lambda w_{(s,t^1)}.{}^\vee W({}^\wedge\lambda T_{(s,e^2)}.{}^\vee w({}^\wedge[{}^\vee T({}^\wedge\lambda x.\text{WALK}({}^\vee x))]))$

The outcome of applying (226) to the intension of (224) can be converted into:

(227) $\lambda w_{(s,t^1)}.[{}^\vee w({}^\wedge\exists u[\text{MAN}(u) \wedge \text{WALK}(u)])$
$\vee\ {}^\vee w({}^\wedge\exists u[\text{WOMAN}(u) \wedge \text{WALK}(u)])]$

And finally, reading (223) is reached when (228), the result of raising the first argument of *believes that*, is applied to the intension of (227) and the intension of j in that order.

(228) $\lambda V_{(s,t^2)}\lambda x_{(s,e)}.{}^\vee V({}^\wedge\lambda p_{(s,t)}.\text{BELIEVE}(p)({}^\vee x))$

The flexible grammar assigns all possible coordination scopes.[63] Leaving aside the readings in which the noun phrase *every student* takes wide scope, sentence (229), for example, is assigned the readings (230) (*i*) through (*iii*):

(229) Mary indicated that every student failed or got a D

(230) (*i*) $\text{INDICATE}(m, \hat{}\forall v[\text{STUDENT}(v) \rightarrow [\text{FAIL}(v) \vee \text{GET-D}(v)]])$

(*ii*) $\text{INDICATE}(m, \hat{}[\forall v[\text{STUDENT}(v) \rightarrow \text{FAIL}(v)] \vee \forall v[\text{STUDENT}(v) \rightarrow \text{GET-D}(v)]])$

(*iii*) $\text{INDICATE}(m, \hat{}\forall v[\text{STUDENT}(v) \rightarrow \text{FAIL}(v)]) \vee \text{INDICATE}(m, \hat{}\forall v[\text{STUDENT}(v) \rightarrow \text{GET-D}(v)])$

Partee and Rooth say that they 'believe that the sentence *Mary indicated that every student failed or got a D* has a reading equivalent to *Mary indicated that every student failed or indicated that every student got a D*, but no reading equivalent to *Mary indicated that every student failed or every student got a D*' (1983, p. 376).

So, while accepting reading (230) (*iii*) (and the 'primary' reading (*i*)), Partee and Rooth reject (*ii*) as a possible reading for (229), though they admit that 'intuitions are far from clear' and that they 'see no non-ad hoc way of capturing this divergence between scope with respect to intensional operators and scope with respect to quantifiers' (1983, p. 376).[64]

Notice, however, that the latter description does not picture the situation correctly. Partee and Rooth's intuitions apparently allow the expression *or* to take scope over *both* the quantifier (*every student*) *and* the intensional operator (*indicated that*) as in (230) (*iii*), but not simply over the quantifier as in (230) (*ii*). Perhaps this pattern becomes less puzzling if it is seen as an instance of a general preference of wide scope-*or* for taking the widest scope in sentences, a preference which can also be detected in the absence of quantifying expressions:

(231) Mary said that John knew that Fred failed or got a D

(232) (*i*) $\text{SAY}(m, \hat{}\text{KNOW}(j, \hat{}[\text{FAIL}(f) \vee \text{GET-D}(f)]))$

(*ii*) $\text{SAY}(m, \hat{}[\text{KNOW}(j, \hat{}\text{FAIL}(f)) \vee \text{KNOW}(j, \hat{}\text{GET-D}(f))])$

(*iii*) $\text{SAY}(m, \hat{}\text{KNOW}(j, \hat{}\text{FAIL}(f))) \vee \text{SAY}(m, \hat{}\text{KNOW}(j, \hat{}\text{GET-D}(f)))$

[63] Except for readings where expressions in conjuncts and disjuncts have wider scope than the conjunctions and disjunctions coordinating them – cf. Section 5.3 below.

[64] Viz., the divergence between (*iii*), where *or* takes scope over the intensional operator *indicated that*, and (*ii*), where it takes scope over the quantifier *every student*.

In addition to the 'standard' reading (232) (*i*), sentence (231) has a widest scope-*or* reading (*iii*) (suggested by the continuation '... but I have forgotten what she said') and an intermediate scope-*or* reading (*ii*) (suggested by the continuation '... but that she had forgotten what he knew'). Observe that the widest scope-*or* reading is more salient, but that the intermediate scope-*or* reading is nonetheless present.[65]

5.3 Structural Restrictions

The grammar defined above is best seen as creating a 'logical space' of quantifier and coordination scope possibilities which is not necessarily exhausted by the facts of particular natural languages. Indeed, in the more and less recent past various structural restrictions on possible readings have been claimed to exist. More in particular, some authors have pointed out the existence of so-called *scope islands*, certain syntactic constituents which restrict the scope of quantifying and coordinating subexpressions to those constituents (so that quantified noun phrases or coordinations occurring inside expressions of island categories cannot have their scope outside of these environments), while others have drawn attention to the existence of what may be called *scope sieves*, syntactic constituents which disallow quantifying and coordinating subexpressions to take scope at the level of those constituents and force them to take scope on a higher or lower level. The flexible grammar should provide the means to impose such structural restrictions on the possibility of readings. In the present section we will show that this holds for scope islands such as relative clauses and coordinate structures and for the sieve category of naked infinitival complements.

5.3.1 Scope Islands

Lakoff (1971,1972) suggests that the general constraints which Ross (1967) proposes for a number of syntactic transformation rules[66] apply as well

[65]The same can be seen with respect to the *Wanda claims that*-sentences in (77) above, where the readings with widest scope-*or* are more prominent than the readings where *or* has narrower scope than *claims that*, but wider scope than *seeks*.

In the particular case of sentence (229), the relative – 'intuitions are far from clear' – unavailability of reading (*ii*) for speakers such as Partee and Rooth is possibly enhanced by the fact that the wide scope-*or* reading of the embedded clause *Every student failed or got a D* entails (i.e., is hidden by) the wide scope-*every* reading (see footnote 32 above).

[66]In varying guises, these constraints have survived successive versions of generative syntax. Since we are only concerned with their empirical consequences for scope phenom-

to the rule of Quantifier Lowering, a rule which Lakoff holds responsible for assigning scope to quantifying expressions. The earliest discussion of these constraints and their relevance for scope phenomena in the framework of Montague grammar is Rodman (1976). Rodman hypothesizes that Ross's Complex NP Constraint (CNPC) and Coordinate Structure Constraint (CSC) can be formulated in this framework in such a way as to make observably correct predictions about quantifier scope relationships in sentence interpretations.

Rodman's paper summarizes the semantic effects of the CNPC in the following way: the quantified head noun phrase of a relative clause always has wider scope than any noun phrase inside that relative clause – i.e., relative clauses constitute scope islands for quantified noun phrases. Sentence (233) is a striking case in point:

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

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

Example (233) has the (strange) interpretation (234) (*i*), but interpretation (234) (*ii*), though describing a situation which is much more normal, is impossible. Note, by the way, that relative clauses are also scope islands for other subexpressions than quantified noun phrases. Whereas sentence (235) can be interpreted as in (236) (*i*), the expression *and* in (235) cannot take scope over *a girl* as in (236) (*ii*):[67]

(235) A girl who lives in Florence and Pisa walks

(236) (*i*) $\exists u[[\text{GIRL}(u) \wedge [\text{LIVE}(f)(u) \wedge \text{LIVE}(p)(u)]] \wedge \text{WALK}(u)]$
(*ii*) $\exists v[[\text{GIRL}(u) \wedge \text{LIVE}(f)(u)] \wedge \text{WALK}(u)]$
$\wedge\ \exists v[[\text{GIRL}(u) \wedge \text{LIVE}(p)(u)] \wedge \text{WALK}(u)]$

Rodman implements the CNPC by introducing a separate class of syntactic variables. These so-called *superscript R variables* are of the form he_i^R, where $i \in \mathbb{N}$. In the course of relative clause formation, moreover, all occurrences of 'normal' syntactic variables in the relative clause are superscripted

ena (but cf. footnote 86 below), we will not go into their precise formulation in particular frameworks.

[67] Variants of (233) and (235) with modifying prepositional phrases instead of relative clauses do have interpretations in which *every* and *and* take wide scope, cf. *A girl in every Italian city walks* and *A girl in Florence and Pisa walks*.

with R, regardless of their subscript. Since Rodman reinterprets the rules of quantification (see (X),i, (XI),i and (XII),i in Section 1.3.2 above) as *requiring* a first occurrence of a non-superscripted syntactic variable he_i in β in order to be applicable,[68] this entails that occurrences of syntactic variables in relative clauses become unavailable for replacement by the noun phrase α in applications of these rules (though such occurrences can still be 'pronominalized', viz., if they belong to 'the remaining occurrences' in β).[69]

The above mechanism disallows wide scope readings for quantified noun phrases occurring in relative clauses. It is, for instance, impossible to obtain reading (234) (i) for example (233), because the expression into which the noun phrase *every Italian city* needs to be quantified in if (233) is to be obtained cannot fail to contain a superscript R variable, so that quantifying-in is blocked:

(237) A girl who lives in he_i^R walks

Rodman observes that the scope of quantified noun phrases also obeys the CSC. Sentence (238), for example, does not possess the scope ambiguities that the PTQ fragment would predict. It has only the two readings specified in (239): 'the quantifiers in the conjoined structure function as a single unit with respect to scope differences vis-à-vis the subject of the sentence' (Rodman (1976), p. 171).[70]

(238) A soldier found every woman and every child

[68]Thus relying on the convention implicit in Montague's work that expressions containing syntactic variables are ill-formed and will somehow be 'thrown out'. An explicit version of Montague's convention, the 'variable principle', has been formulated by Janssen (1986, Part 2, pp. 137–173).

[69]Another desirable consequence of Rodman's proposal is that it rightly blocks the relativization of elements of relative clauses: **The fish which John has dated a woman who loves walks*.

[70]That is, if the PTQ fragment were extended with a rule which permits the construction of conjoined term phrases $[_{\mathrm{T}}[_{\mathrm{T}}\alpha]$ *and* $[_{\mathrm{T}}\beta]]$ and interprets these as $\lambda P.[\alpha'(P) \wedge \beta'(P)]$. Rodman says that such a fragment would predict example (238) to be six ways ambiguous, but this is wrong: all interpretations of a sentence T' *found* T'' *and* T''' (where T', T'' and T''' are 'simple' term phrases, cf. footnote 59 above) can be obtained by quantifying in

(a) T' and T''' into the sentence he_i *found* T'' *and* he_j;

(b) T' and T'' into the sentence he_i *found* he_j *and* T''';

(c) T', T'' and T''' into the sentence he_i *found* he_j *and* he_k; and

(d) T' and T'' *and* T''' into the sentence he_i *found* he_j.

This yields the following twelve potentially non-equivalent derivations (where the sentences

(239) (*i*) $\exists u[\text{SOLDIER}(u) \wedge [\forall v[\text{WOMAN}(v) \rightarrow \text{FIND}(v)(u)] \wedge \forall w[\text{CHILD}(w) \rightarrow \text{FIND}(w)(u)]]]$

(*ii*) $\forall v[\text{WOMAN}(v) \rightarrow \exists u[\text{SOLDIER}(u) \wedge \text{FIND}(v)(u)]] \wedge \forall w[\text{CHILD}(w) \rightarrow \exists u[\text{SOLDIER}(u) \wedge \text{FIND}(w)(u)]]$

Apparently, quantifying-in into a coordinate structure is not allowed. The semantically simpler sentence (240) may serve to illustrate the point more clearly:

(240) Mary finds John and every unicorn

Enriched with a rule for term phrase conjunction (cf. footnote 70), the PTQ grammar will assign two interpretations to this sentence:

(241) (*i*) $\text{FIND}(j)(m) \wedge \forall u[\text{UNICORN}(u) \rightarrow \text{FIND}(u)(m)]$

(*ii*) $\forall u[\text{UNICORN}(u) \rightarrow [\text{FIND}(j)(m) \wedge \text{FIND}(u)(m)]]$

---

mentioned in (*a*), (*b*), (*c*) and (*d*) are abbreviated as A, B, C and D, respectively):

1.	$T_i' + (T_j''' + A)$	*5.*	$T_i' + (T_j'' + (T_k''' + C))$	*9.*	$T_k''' + (T_i' + (T_j'' + C))$
2.	$T_j''' + (T_i' + A)$	*6.*	$T_i' + (T_k''' + (T_j'' + C))$	*10.*	$T_k''' + (T_j'' + (T_i' + C))$
3.	$T_i' + (T_j'' + B)$	*7.*	$T_j'' + (T_i' + (T_k''' + C))$	*11.*	$T_i' + (T'' \textit{ and } T_j''' + D)$
4.	$T_j'' + (T_i' + B)$	*8.*	$T_j'' + (T_k''' + (T_i' + C))$	*12.*	$T'' \textit{ and } T_j''' + (T_i' + D)$

Since in the particular case of sentence (238) both T'' and T''' are of the form *every CN*, the derivations *5* and *6* as well as the derivations *8* and *10* receive equivalent translations. Of the remaining ten translations, only *11* and *12* supply reasonable interpretations for (238), viz., (239) (*i*) and (239) (*ii*), respectively. Derivation *1*, for example, generates a translation which can be reduced to:

$$\exists u[\text{SOLDIER}(u) \wedge [\forall w[\text{CHILD}(w) \rightarrow \text{FIND}(w)(u)] \wedge [\exists w\, \text{CHILD}(w) \rightarrow \forall v[\text{WOMAN}(v) \rightarrow \text{FIND}(v)(u)]]]]$$

This formula expresses that there is a soldier who found every child and, *if there are children*, every woman. Parallel problems arise with the translations furnished by *3*, *5/6*, *7* and *9* – see also translation (241) (*ii*) of sentence (240). Moreover, derivation *2* results in the following translation:

$$\forall w[\text{CHILD}(w) \rightarrow \exists u[\text{SOLDIER}(u) \wedge [\text{FIND}(w)(u) \wedge \forall v[\text{WOMAN}(v) \rightarrow \text{FIND}(v)(u)]]]]$$

'Every child was found by a soldier who found every woman', which is problematic for similar reasons as the translations produced by the derivations *4* ('Every woman was found by a soldier who found every child') and *8/10* ('For every woman and every child there is a soldier who found them both').

Interpretation (241) (*i*) can be obtained by constructing the sentence without applying rules of quantification. In order to get (241) (*ii*), one has to use some rule (X),*i*, quantifying in the term phrase *every unicorn* into the sentence *Mary finds John and* he_i.

However, while (241) (*i*) represents a reading of (240), formula (241) (*ii*) is an incorrect translation of the sentence, as can be seen from the equivalence of the following formulas:

(242) $\forall u[\text{UNICORN}(u) \rightarrow [\text{FIND}(j)(m) \wedge \text{FIND}(u)(m)]]$
$\forall u[[\text{UNICORN}(u) \rightarrow \text{FIND}(j)(m)] \wedge [\text{UNICORN}(u) \rightarrow \text{FIND}(u)(m)]]$
$\forall u[\text{UNICORN}(u) \rightarrow \text{FIND}(j)(m)] \wedge \forall u[\text{UNICORN}(u) \rightarrow \text{FIND}(u)(m)]$
$[\exists u\, \text{UNICORN}(u) \rightarrow \text{FIND}(j)(m)] \wedge \forall u[\text{UNICORN}(u) \rightarrow \text{FIND}(u)(m)]$

Translation (241) (*ii*) apparently expresses: Mary finds John if there are unicorns, and for every unicorn it holds that Mary finds it – a truth in every possible world without unicorns, whether or not Mary finds John. This is wrong, of course: sentence (240) does not express that the finding of John by Mary is dependent on the existence of unicorns.

The impossibility of quantifying-in into coordinate structures extends to sentences containing other conjoined expressions than term phrases. Thus example (243) does have reading (244), in which *every town* and *every village* take scope over *an inspector*:

(243) An inspector visited every town and phoned every village

(244) $\forall v[\text{TOWN}(v) \rightarrow \exists u[\text{INSPECTOR}(u) \wedge \text{VISIT}(v)(u)]]$
$\wedge \forall w[\text{VILLAGE}(w) \rightarrow \exists u[\text{INSPECTOR}(u) \wedge \text{PHONE}(w)(u)]]$

But reading (244) cannot be obtained by allowing rules of quantification to apply to syntactic variables occurring in coordinate intransitive verb phrases, for observe that the direct object noun phrases take scope over the subject noun phrase in (244), but not over the expression *and*.[71] Rather, what

[71] Accordingly, (244) is not among the readings assigned in the PTQ fragment: constructing (243) without rules of quantification leads to the following reading:

$\exists u[\text{INSPECTOR}(u) \wedge [\forall v[\text{TOWN}(v) \rightarrow \text{VISIT}(v)(u)] \wedge \forall w[\text{VILLAGE}(w) \rightarrow \text{PHONE}(w)(u)]]]$

Moreover, let T', T'' and T''' represent *an inspector*, *every town* and *every village*, respectively. Then all other PTQ interpretations of (243) are obtained by quantifying in
(*a*) T' and T''' into the sentence he_i *visited* T'' *and phoned* he_j;
(*b*) T' and T'' into the sentence he_i *visited* he_j *and phoned* T'''; and

(244) shows is the necessity of assigning wide scope readings to coordinating expressions.[72]

In view of these facts (and parallel data involving the non-relativizability of elements of coordinate expressions), Rodman proposes a reformulation of the PTQ rules for conjunction and disjunction of terms phrases, intransitive verb phrases and sentences: in the process of coordination, all occurrences of syntactic variables are replaced by superscript R variables, regardless of their subscript. Together with the reinterpretation of the rules of quantification sketched above, this reformulation entails a ban on quantifying in into coordinate structure (though, again, pronominalization remains possible).

The applicability of the CNPC to quantifier scope relationships has been studied extensively by Cooper (1975; 1979; 1983). Cooper (1975, p. 320, footnote 6) points out that examples like (245) do permit a reading in which the noun phrase *a producer I know* has a wide scope interpretation (cf. (246) (*ii*)), such in violation of the CNPC:

(245) Mary dates every man who knows a producer I know

(246) (*i*) $\forall v[[\text{MAN}(v) \wedge \exists u[[\text{PRODUCER}(u) \wedge \text{KNOW}(u)(i)] \wedge \text{KNOW}(u)(v)]] \rightarrow \text{DATE}(v)(m)]$

(*ii*) $\exists u[[\text{PRODUCER}(u) \wedge \text{KNOW}(u)(i)] \wedge \forall v[[\text{MAN}(v) \wedge \text{KNOW}(u)(v)] \rightarrow \text{DATE}(v)(m)]]$

The most elaborate discussion of putative counterexamples to the claim that quantifier scope is constrained by the CNPC is Cooper (1979). But here the author concludes: 'While the judgements for these sentences are somewhat

(*c*) T', T'' and T''' into the sentence *he$_i$ visited he$_j$ and phoned he$_k$*.

Let A, B and C abbreviate the sentences mentioned in (*a*), (*b*) and (*c*), respectively. The options (*a*), (*b*) and (*c*) give rise to ten potentially non-equivalent derivations which are structurally identical to the derivations listed as *1* through *10* in footnote 70 above. Again, the translations of (the counterparts of) both *5* and *6* and *8* and *10* are equivalent (since both T'' and T''' are of the form *every CN*), and in practically the same way as the interpretations of the derivations *1* through *10*, the remaining eight interpretations can all be shown to be non-readings of sentence (243).

[72] In the fragment of Section 4 above, reading (244) is obtained by combining the basic translations of the deteminers and the nouns with the derived translations which result from subjecting the basic translations of the verbs to the following type shift:

$$((s,e),((s,e),t)) \Rightarrow ((s,e),((s,e^2),t)) \Rightarrow ((s,e^2),((s,e^2),t))$$

subtle I believe that there is enough evidence to cast serious suspicion on them as counterexamples to the claim that quantification obeys the CNPC' (p. 149). In the case of sentence (245), for instance, there seems to be no reason *a priori* to suppose that the sentence is ambiguous between a wide scope-*a* and a narrow scope-*a* reading, since the narrow scope reading (246) (*i*) will always be true if the wide scope reading (246) (*ii*) is true. Therefore, 'what appears to be a preferred wide scope reading we may take to be a narrow scope reading which is restricted by our knowledge of the world' (p. 142). Cooper (1979) proposes to account for the CNPC in terms of a condition on NP storage: 'although a complete NP translation can be stored, an NP cannot be translated with a constituent NP stored'.[73] Finally, in Cooper (1983) – which has this condition built-in into its rule of NP storage[74] – it is claimed that an answer to the question whether the island constrains apply to wide scope quantification requires 'a detailed discussion of the meaning of particular noun phrases' (p. 136).

Such a discussion is given in Fodor and Sag (1982). Fodor and Sag argue convincingly that there is 'evidence for a semantic ambiguity in indefinite noun phrases, over and above any scope ambiguities in appropriate contexts' (1982, p. 355).[75] They note that Cooper's idea that the wide scope-*a* reading (246) (*ii*) of sentence (245) is in fact the weaker narrow scope-*a* reading restricted by knowledge of the world will not work for (247). This example does have the wide scope-*a* reading (248) (*ii*), but the narrow scope-*a* reading (248) (*i*) is in fact *stronger* than (248) (*ii*):[76]

(247) Mary dates at most three men who know a producer I know

(248) (*i*) $\exists_{\leq 3} v[[\text{MAN}(v) \wedge \exists u[[\text{PRODUCER}(u) \wedge \text{KNOW}(u)(i)] \wedge \text{KNOW}(u)(v)]] \wedge \text{DATE}(v)(m)]$

(*ii*) $\exists u[[\text{PRODUCER}(u) \wedge \text{KNOW}(u)(i)] \wedge \exists_{\leq 3} v[[\text{MAN}(v) \wedge \text{KNOW}(u)(v)] \wedge \text{DATE}(v)(m)]]$

[73] Which comes in handy anyway, because 'allowing double storage of this type would in fact have disastrous consequences in the system proposed in Cooper (1975)' (Cooper (1979), pp. 157–158). Cf. footnote 15 above.

[74] See (25) and footnote 11 above.

[75] For simplicity, Fodor and Sag limit their arguments to noun phrases containing the singular indefinite article *a*, but they believe that a comparable ambiguity can be demonstrated for numerical determiners.

[76] Another case in point is *Mary dates exactly half the men who know a producer I know*, where there is no entailment relation between the two readings.

By contrast, a sentence like *Mary dates at most three men who know every producer I know* does not have an island-escaping wide scope-*every* reading (even though the entailment relation runs in the right direction).

Fodor and Sag account for this contrast by assigning both a *referential* and a *quantifier* interpretation to indefinite noun phrases.[77] On its quantifier interpretation, an indefinite noun phrase exhibits the same, island-observing, scope ambiguities as purely (i.e., exclusively) quantified expressions such as *every producer I know*. The referential interpretation of an indefinite noun phrase, on the other hand, gives rise to sentence readings which are more or less equivalent[78] to a widest scope quantifier reading. It is the referential interpretation which is responsible for the island-'escaping' behaviour of indefinite noun phrases: 'a referential phrase doesn't so much escape from a scope island as shine right through it; and it also shines right through any scoped elements in the sentence' (Fodor and Sag (1982), p. 375). This explains why an indefinite which manages to 'escape' from a scope island has always maximally wide scope with respect to any quantifiers or logical operators outside the island.[79]

---

[77]They note that the availability of the referential interpretation is favoured by the descriptive richness of the indefinite noun phrase. Thus *a producer I know* is more likely to be understood referentially than *a producer*.

[78]Strictly speaking, a referential reading only entails a widest scope existential quantifier reading: the truth of the existential reading guarantees that *some* referential statement must be true, but it does not guarantee the truth of any particular referential statement.

[79]This observation is contested in Abusch (1993). Fodor and Sag mention another advantage of assuming a separate referential interpretation of indefinite noun phrases: in Sag (1977) and Williams (1977) it is observed that VP deletion is subject to a semantic identity condition, and in particular that a verb phrase cannot be deleted if its antecedent contains a quantified noun phrase which has wider scope than the verb phrase. Thus, they claim that example (*a*) is unacceptable if *every town* takes scope over *an inspector*:

(*a*) *An inspector visited every town. An idler did too.*

However, indefinite noun phrases are not subject to this condition. Example (*b*) can easily be understood as being about a particular teacher:

(*b*) *Every girl likes a teacher I know. Every boy does too.*

The general condition of Sag (1977) is, approximately, that a verb phrase may be deleted only if its logical translation is an alphabetic variant of an expression in the logical translation of the surrounding discourse.

While the wide scope-*a* interpretation of (*a*) satisfies this condition, the wide scope-*every* interpretation does not: in Sag's treatment, the wide scope-*every* translation of the deleted VP will contain the universally quantified variable but not its quantifier, so that this variable occurs freely in the translation. Since Sag's translation principles do not

On their referential interpretation, indefinite noun phrases are analyzed as indexical expressions. Their denotation is determined both by their descriptive content and by the context of utterance, which includes the psychological state of the speaker. According to Fodor and Sag, a referential indefinite noun phrase $[_{\mathrm{T}}[_{\mathrm{Det}}\, a][_{\mathrm{CN}} B]]$ denotes the individual the speaker 'has in mind' – the individual the speaker intends to refer to at the time of the utterance of the indefinite noun phrase –, as long as the descriptive content B is true of that individual. Otherwise, it denotes †, the 'completely alien entity' which represents the 'undefined' value of a function. So, truth of sentences is relativized to contexts of utterance c which, among other things, specify a 'referentially intended' individual d_c.[80] As a result of this

guarantee the absolute identity of this variable, the antecedent and the deleted VP will not be alphabetic variants. Given a quantifier interpretation of indefinites, reverse predictions are made for (*b*): the wide scope-*every* interpretation of (*b*) satisfies Sag's condition, but the wide scope-*a* interpretation does not. However, on the assumption that indefinites can be interpreted as referential terms (which do not bind variables), there is a 'wide scope-*a*' interpretation of (*b*) satisfying the condition, which, consequently, can be retained.

Fodor and Sag rightly remark that all this 'depends heavily on the details of the logical representation language for natural language sentences' (1982, p. 377). This aspect can be improved in the flexible set-up of Section 4 above: the fact that – modulo referential interpretations – a verb phrase cannot be deleted if its antecedent contains a quantified noun phrase which has wider scope than the verb phrase will simply ensue from supplementing Fodor and Sag's analysis of indefinites with the requirement that the interpretation of an antecedent of a deleted VP be of the basic type $((s,e),t)$.

In his discourse theory of VP deletion, Prüst (1992) does not accept the non-wide scope generalization of Sag (1977) and Williams (1977) – on account of counterexamples, however, which all involve indefinite or numerical noun phrases, and hence, plausibly, referential interpretations (Prüst (1992), pp. 81, 83, 97, 99). Given flexible interpretation (and referential indefinites), the generalization can be obtained in Prüst's (extensional) theory by translating the anaphoric VP *does* as a unification variable $\underline{\mathrm{do}}$ of the lowest VP type (e,t). As this variable only unifies with antecedent interpretations of the same type, wide scope readings of non-referential noun phrases in the antecedent will be excluded.

[80] Recasting the formal semantics of Fodor and Sag in an intensional and compositional format, we add a variable binding operator O to **IL** which, together with a variable v of type e and an expression φ of type t, forms an expression $\mathrm{O}v\,\varphi$ of type e. We assume that the reference failure entity † is a member of the domain of individuals E. The extension of terms α is determined given a model m, a world w, an assignment g, and a context of utterance c, where c is an ordered pair $\langle d_c, w_c\rangle$ consisting of a referentially intended individual d_c and a possible world w_c (the latter is needed because the descriptive content B of a referential indefinite $[_{\mathrm{T}}[_{\mathrm{Det}}\, a][_{\mathrm{CN}} B]]$ is required to be true of d_c in the world of the context of utterance, not just in any old world). Further, we add c as a parameter to the definition of the extensions of all familiar terms, and stipulate that

$|\mathrm{O}v\,\varphi|^{m,w,g,c} = d_c$ if $|\varphi|^{m,w_c,g[v\to d_c],c} = 1$; otherwise $|\mathrm{O}v\,\varphi|^{m,w,g,c} = \dagger$.

analysis,[81] the denotation of a referential indefinite depends only on the context of utterance and not on the sentential context in which the phrase appears. Referential indefinites are thus insensitive to scoped elements in the sentence.

The upshot of the above discussion is, then, that the CNPC and the CSC do govern scope phenomena: relative clauses and coordinations constitute scope islands for quantified (non-referential) noun phrases and coordinating expressions.[82] So, how can these structural restrictions be incorporated into

---

The determiner $[_{\mathrm{Det}}\, a]$ is translated as $\lambda P_{(s,((s,e),t))}.O v\, {}^{\vee}P({}^{\wedge}v)$ of type $((s,((s,e),t)),e)$. This results in a reduced translation $O v\, \beta({}^{\wedge}v)$ of type e for referential indefinite noun phrases $[_{\mathrm{T}}[_{\mathrm{Det}}\, a][_{\mathrm{CN}} B]]$ (β represents the translation of the common noun B). Hence, reduced sentence translations in which a noun phrase $[_{\mathrm{T}}[_{\mathrm{Det}}\, a][_{\mathrm{CN}} B]]$ is interpreted referentially are of the form $\alpha[v := O v\, \beta({}^{\wedge}v)]$, where $\exists v[\beta({}^{\wedge}v) \wedge \alpha]$ expresses the widest scope quantifier 'reading' of the sentence. The insensitivity of referential indefinites to other elements in the sentence is reflected in the FACT that for all τ, v, φ, m, w, g and c:

$|\tau[v := O v\, \varphi]|^{m,w,g,c} = |\tau|^{m,w,g[v\to d],c}$, where $d = |O v\, \varphi|^{m,w,g,c}$

(which holds since for all w, w': $|O v\, \varphi|^{m,w,g,c} = |O v\, \varphi|^{m,w',g,c}$, viz., d_c if $|\varphi|^{m,w_c,g[v\to d_c],c} = 1$, and $\dagger$ otherwise). Note that, consequently, a referential reading $\alpha[v := O v\, \beta({}^{\wedge}v)]$ entails the widest scope existential quantifier reading $\exists v[\beta({}^{\wedge}v) \wedge \alpha]$. Skirting possible difficulties due to reference failure, we restrict our attention to situations where the speaker correctly describes the individual (s)he has in mind: $|\beta({}^{\wedge}v)|^{m,w_c,g[v\to d_c^*],c} = 1$. Then, if $|\alpha[v := O v\, \beta({}^{\wedge}v)]|^{m,w_c,g,c} = 1$, we can use the FACT to conclude that $|\alpha|^{m,w_c,g[v\to d],c} = 1$ for $d = |O v\, \varphi|^{m,w_c,g,c}$, i.e., d_c. So $|\beta({}^{\wedge}v) \wedge \alpha|^{m,w_c,g[v\to d_c],c} = 1 = |\exists v[\beta({}^{\wedge}v) \wedge \alpha]|^{m,w_c,g,c}$.

[81] A drawback of Fodor and Sag's analysis is that it permits only one referentially intended d_c per utterance situation. Thus on its doubly referential interpretation, an utterance like *A producer I know likes an actress I know* will only be true if the individual the speaker has in mind is a producer and an actress who is known by the speaker and who likes herself. This can be avoided by developing a theory of smaller – subsentential – utterance situations, or (perhaps less laboriously) by assuming (*i*) that d_c is a sequence of individuals $\langle d_c^1, \ldots, d_c^n \rangle$; (*ii*) that $|O v_i\, \varphi|^{m,w,g,c} = d_c^i$ if $|\varphi|^{m,w_c,g[v\to d_c^i],c} = 1$, and $\dagger$ otherwise; (*iii*) that determiners are subscripted (following the practice of dynamic Montague grammar – see Section 6 below); and (*iv*) that $[_{\mathrm{Det}}\, a_i]$ is translated as $\lambda P_{(s,((s,e),t))}.O v_i\, {}^{\vee}P({}^{\wedge}v_i)$.

[82] From the outset, the applicability of the CNPC and the CSC to the binding of pronouns – the second main task of rules of quantification – has been doubted. Thus, Rodman (1976) notes that in a sentence like *A soldier found every student and shot him*, the pronoun *him* can be bound by *every student*. In view of such violations of the CSC, Rodman concludes that 'while we must maintain strict control over relativization by means of the superscript R variables, the controls over quantification, which I had originally hoped would parallel all the way down the line the controls over relativization must in fact be weakened somewhat' (p. 173). Similar sentences lead Cooper (1979) to the claim that the CSC 'is not completely obeyed by the quantifying-in mechanism' (p. 155). Cooper (1979) also discusses examples with anaphoric pronouns which seem to violate the CNPC, such as the (donkey) sentence *Every girl who ate a steak enjoyed it*, if *a steak* is assigned a non-referential interpretation. Considering this example, Fodor and Sag (1982) announce that they 'will have nothing to say about [...] the existence of anaphoric relations be-

the flexible grammar?

As it happens, the fragment of Section 4 does observe the CSC. For example, the sentence *Mary finds John and every unicorn* is only assigned translations which reduce to (249) (*i*), and none of its translations is equivalent to (249) (*ii*) (see the Appendix):

(249) (*i*) $\text{FIND}(j)(m) \wedge \forall u[\text{UNICORN}(u) \rightarrow \text{FIND}(u)(m)]$
(*ii*) $\forall u[\text{UNICORN}(u) \rightarrow [\text{FIND}(j)(m) \wedge \text{FIND}(u)(m)]]$

It is important to note, however, that this result is a by-product of our having separate rules of conjunction and disjunction in which the expressions *or* and *and* are treated syncategorematically (a treatment which facilitated comparison with the fragments of PTQ and Partee and Rooth (1983)).

As an alternative, consider an analysis which polymorphically (i.e., for any finite sequence of types $\vec{a} = a_1, \ldots, a_n$) assigns the respective basic translations in (250) to *or* and *and*, and puts these coordinating expressions into a syntactic category CO which figures in rule (251):[83]

(250) $\lambda v_{n+1,(s,(\vec{a},t))} \lambda v_{n+2,(s,(\vec{a},t))} \lambda \vec{x}_{\vec{a}} . [{}^{\vee} v_{n+1}(\vec{x}) \vee {}^{\vee} v_{n+2}(\vec{x})]$
$\lambda v_{n+1,(s,(\vec{a},t))} \lambda v_{n+2,(s,(\vec{a},t))} \lambda \vec{x}_{\vec{a}} . [{}^{\vee} v_{n+1}(\vec{x}) \wedge {}^{\vee} v_{n+2}(\vec{x})]$

(251) For $C \in \{\text{S}, \text{IV}, \text{T}, \text{TV}, \text{CN}\}$:
$[_C \alpha] + [_{\text{CO}} \beta] + [_C \gamma] \rightarrow \delta$, where $\delta = [_C [_C \alpha][_{\text{CO}} \beta][_C \gamma]]$
$Tr_B(\delta) = \{\text{FA}_2(\alpha', \beta', \gamma') \mid \alpha' \in Tr(\alpha),\ \beta' \in Tr(\beta) \text{ and } \gamma' \in Tr(\gamma)\}$

On this categorematic analysis of coordination, the non-interpretation expressed by (249) (*ii*) will become available after all: an equivalent translation results when the first argument of the basic translation of *finds* is raised and translation (254) is employed, a derived translation of *and* which results from applying the type shifts in (253) to the basic translation (252) which is of type $((s, e^2), ((s, e^2), e^2))$:

(252) $\lambda T_1 \lambda T_2 \lambda P . [{}^{\vee} T_1(P) \wedge {}^{\vee} T_2(P)]$

---

tween noun phrases and pronouns that are apparently not within their scope', though they do note that this 'problem cannot be solved simply by acknowledging a referential interpretation for indefinites.' (p. 394, footnote 10). We will relegate the treatment of 'anaphoric relations between noun phrases and pronouns that are apparently not within their scope' to the theory of dynamic Montague grammar (see Section 6 below), which has been developed with the intention of handling precisely these phenomena.

[83] The semantic operation FA_2 is defined for β' of type $((s,a),((s,c),b))$ and α' and γ' of respective types a and c: $\text{FA}_2(\alpha', \beta', \gamma') = \beta'(\hat{}\alpha')(\hat{}\gamma')$.

(253) $((s,e^2),((s,e^2),e^2)) \Rightarrow ((s,e),((s,e^2),e^2)) \Rightarrow ((s,e),((s,e),e^2))$
$\Rightarrow ((s,e),((s,e^2),e^2)))$

(254) $\lambda x_1 \lambda T_2 \lambda P.{}^{\vee}T_2({}^{\wedge}\lambda x_2.[{}^{\vee}P(x_1) \wedge {}^{\vee}P(x_2)])$

For lack of relative clauses, the flexible fragment vacuously meets the CNPC. But if an (overly simple)[84] treatment of relative clauses is assumed, according to which relative pronouns such as *who* are assigned the category RP and the basic translation specified in (255), and rule (256) is added, then the sentence *A girl who lives in every Italian city walks* will incorrectly receive a translation which reduces to (257):

(255) $\lambda P_{(s,((s,e),t))} \lambda Q_{(s,((s,e),t))} \lambda x_{(s,e)}.[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)]$

(256) $[_{\text{CN}}\alpha] + [_{\text{RP}}\beta] + [_{\text{IV}}\gamma] \rightarrow \delta$, where $\delta = [_{\text{CN}}[_{\text{CN}}\alpha][_{\text{RP}}\beta][_{\text{IV}}\gamma]]$
$Tr_B(\delta) = \{\text{FA}_2(\alpha', \beta', \gamma') \mid \alpha' \in Tr(\alpha),\ \beta' \in Tr(\beta) \text{ and } \gamma' \in Tr(\gamma)\}$

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

Pretend, for convenience, that *lives in* is an extensional transitive verb which translates as $\lambda x \lambda y.\text{LIVE}({}^{\vee}x)({}^{\vee}y)$ of type $((s,e),((s,e),t))$. Then (257) is the outcome of using the translations which result from applying the following type shifts to the basic translations of *lives in*, *who*, *a* and *walks*, respectively:

(258) $((s,e),((s,e),t)) \Rightarrow ((s,e),((s,e),t^2)) \Rightarrow ((s,e^2),((s,e),t^2))$
$((s,e^1),((s,e^1),e^1)) \Rightarrow ((s,e^1),((s,e^1),e^3)) \Rightarrow ((s,e^3),((s,e^1),e^3))$
$((s,e^1),((s,e^1),t)) \Rightarrow ((s,e^3),((s,e^1),t))$
$((s,e),t) \Rightarrow ((s,e^2),t)$

This means that there is a problem. As fas as the CSC is concerned, one could stick to the syncategorematic treatment of coordination, but of the two analyses sketched above, the categorematic approach is to be preferred.[85] However, this choice does not necessarily commit one to a *semantics* in which the CSC fails to hold. Failure of the CSC essentially is the result of allowing type-shifting rules to apply to translations of coordinating expressions. Once translations of expressions of the category CO are excluded

[84] Rule (256) allows only relative clauses of which the subject is relativized.

[85] If only for the conceptual reason that it is more intuitively compositional. Cf. Janssen (1986), Part 1, pp. 13–14.

from the rules for deriving translations, it will be impossible to obtain interpretations where expressions in conjuncts and disjuncts have wider scope than the conjunctions and disjunctions coordinating them: it is not hard to see that every interpretation assigned in the fragment of Section 4 can also be assigned categorematically without applying type shifts to translations of coordinating expressions, and *vice versa*. So the CSC can be implemented in a straightforward and natural manner, viz., as a constraint on the translation set of coordinating expressions.

The CNPC can be accounted for in the same way. Notice that application of type shifts to *who* is essential for getting the wrong reading (257). So the remedy is the same: once translations of expressions of the category RP are excluded from the application of type-shifting rules, the fact that relative clauses are barriers for quantification and coordination scope will obtain.

While this move turns both coordinations and relative clauses into scope islands, there is also a semantic difference between the two constructions. Though we do want to prevent quantified noun phrases from shifting out of coordinate structures, we do not want to prevent the coordinate structure itself, including the quantified noun phrases it contains, from taking wide scope over other constituents. This is accounted for by the fact that coordinating expressions are assigned polymorphic sets of basic translations, which enables coordinations to take wide scope themselves – they are 'floating' scope islands, so to speak. Relative pronouns, on the other hand, have only the basic translation of *and* which is specified in (255). As a consequence, relative clauses do not 'float', which is as it should be.[86] This can

[86] Notice that the proposed restriction on the application of type shifts entails that translations of relative clauses are always of type $((s,e),t)$, while translations of conjuncts and disjuncts invariably have the same conjoinable type $(\vec{a},t)$. Perhaps this can be related to the behaviour of these constructions with respect to syntactic unbounded dependencies such as constituent question formation. Thus, it is impossible to extract a phrase from a relative clause, cf. **Which Italian city does a girl who lives in walk?* In the analysis of Gazdar, Klein, Pullum and Sag (1985, p. 230), the type of translations of gap-containing expressions of category C differs from the type of translations assigned to their gapless namesakes: a translation of an expression of category C which contains a gap of category C' is assigned the type $((s,\text{TYPE}(C')),\text{TYPE}(C))$. Maybe, then, extraction from relative clauses is disallowed since it leads to translations of an inadmissible type. As for coordinations, observe that it is impossible to extract a phrase from one conjunct (disjunct), witness **Which singer does Mary love and (or) John walk?*, but that extraction 'across-the-board', viz., from both conjuncts (disjuncts), is okay: *Which singer does Mary love and (or) John hate?* By itself, the type $(\vec{a},t)$ does not exclude that coordinations contain gaps, but if they do, then both coordinated constituents must contain them, because the presence of only one gap would lead to different types of the coordinated translations.

be illustrated with the following minimal pair:

(259) Every man and walker talks

(260) Every man who walks talks

We have seen above (cf. example (209)) that a sentence like (259) gets the two readings listed in (261). Example (260), by contrast, is unambigously interpreted as (261) (*i*):

(261) (*i*) $\forall v[[\text{MAN}(v) \wedge \text{WALK}(v)] \rightarrow \text{TALK}(v)]$
(*ii*) $\forall v[\text{MAN}(v) \rightarrow \text{TALK}(v)] \wedge \forall v[\text{WALK}(v) \rightarrow \text{TALK}(v)]$

5.3.2 Scope Sieves

Another kind of structural restriction on the availability of scope readings is the phenomenon of scope sieves: quantified noun phrases and coordinating expressions 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[87] by Asher and Bonevac (1985; 1987), Muskens (1989) and Van der Does (1990; 1992). These authors observe that a sentence like (262), which according to their analysis contains a sentence-like complement of the perception verb *sees* (see (263)), cannot be interpreted as in (264) (*i*). The embedded subject must have scope in the matrix sentence, cf. (264) (*ii*):

(262) John sees every man swim

(263) $[_{\text{S}}\textit{John}[_{\text{VP}}\textit{sees}[`_{\text{S}}\textit{every man swim}]]]$

(264) (*i*) $\text{SEE}(j, {}^\wedge\forall v[\text{MAN}(v) \rightarrow \text{SWIM}(v)])$
(*ii*) $\forall v[\text{MAN}(v) \rightarrow \text{SEE}(j, {}^\wedge\text{SWIM}(v)])$

[87] And disputed: Akmajian (1977) and Williams (1983) postulate the syntactic structure $[_{\text{S}}\textit{John}[_{\text{VP}}\textit{sees}[_{\text{NP}}\textit{every man}][_{\text{VP}}\textit{swim}]]]$ for example (262), in which the naked infinitival 'complement' *every man swim* is not even a constituent. Given this structure, the scope facts discussed here can be accounted for by combining the 'value raising' proposal for naked infinitival verbal forms made in the present section with a lexical semantics for *sees* along the lines of Dowty's (1985) proposal for raising verbs: $\lambda x_{(s,e)}\lambda P_{(s,e^1)}\lambda y_{(s,e)}.\text{SEE}({}^\wedge[{}^\vee P(x)])({}^\vee y))$. Note, by the way, that Akmajian and Williams's analysis entails that a sentence like *John sees Mary walk and Bill talk* is an instance of non-constituent coordination.

The correct readings cannot be characterized by saying that perception verbs may not take scope over quantified noun phrases at all. For example, in the perfectly acceptable *de dicto* reading (266) (*i*) of (265), the verb *sees* has scope over the determiner *a*. Sentence (265) also has a correct *de re* reading, viz., (266) (*ii*). The intermediate interpretation (266) (*iii*), however, in which *a pen* is interpreted *de dicto* with respect to *sees*, but *de re* as regards *seek*, is not available:

(265) John sees Mary seek a pen

(266) (*i*) $\text{SEE}(j, {}^\wedge\text{SEEK}({}^\wedge\lambda P.\exists u[\text{PEN}(u) \wedge {}^\vee P({}^\wedge u)])(m))$
(*ii*) $\exists u[\text{PEN}(u) \wedge \text{SEE}(j, {}^\wedge\text{SEEK}({}^\wedge\lambda P.{}^\vee P({}^\wedge u))(m))]$
(*iii*) $\text{SEE}(j, {}^\wedge\exists u[\text{PEN}(u) \wedge \text{SEEK}({}^\wedge\lambda P.{}^\vee P({}^\wedge u))(m)])$

Given type flexibility, these facts can be accounted for as follows. The verb *sees* is straightforwardly assigned the basic translation in (267). For the naked infinitival form of verbs of which the tensed form has the basic translation α of type $(\vec{a}, t)$ we assume the basic 'value-raised' translation of type $(\vec{a}, t^2)$ which is given in (268):[88]

(267) $\lambda p_{(s,t)}\lambda x_{(s,e)}.\text{SEE}(p)({}^\vee x)$

(268) $\lambda\vec{x}_{\vec{a}}\lambda w_{(s,((s,t),t))}.{}^\vee w({}^\wedge[\alpha(\vec{x})])$

The naked infinitival form *seek*, for example, is assigned the basic translation (269) of type $((s, e^2), ((s, e), t^2))$.

(269) $\lambda T_{(s,e^2)}\lambda y_{(s,e)}\lambda w_{(s,t^1)}.{}^\vee w({}^\wedge\text{SEEK}(\text{T})({}^\vee y))$

If (269) is combined with the basic translations of *a pen* and *Mary*, respectively, and the result is combined with (270), the term obtained by raising the first argument of the basic translation of *sees*, and the basic translation j of *John*, then reading (266) (*i*) is generated.

(270) $\lambda R_{(s,t^2)}\lambda x_{(s,e)}.{}^\vee R({}^\wedge\lambda p.\text{SEE}(p)({}^\vee x))$

[88] Observe that $(\vec{a}, t^2) = (\vec{a}, ((s, ((s, t), t)), t))$. Maybe this is the right translation for verbal stems in general, where the usual $(\vec{a}, t)$-type translation results from morphologically composing the last $(s, ((s, t), t))$-argument with the interpretation of aspects and tenses such as *to*, *-ed* and *-ing*.

If, in addition, the first argument of the naked infinitive translation is lowered and raised again, we get (266) (*ii*). It is impossible to obtain (266) (*iii*).

This analysis also immediately accounts for the principles of conjunction distribution and disjunction distribution which have been claimed to hold for naked infinitival complements (see Barwise (1981), Barwise and Perry (1983), and the aforementioned authors). For example, sentence (271) is assigned exactly one reading, namely (273), which is also the only reading of sentence (272) and which can be constructed using (270) and the basic translations of the other expressions. Due to the 'higher order' basic translations of the naked infinitives *walk* and *talk*, the conjoined complement *Mary walk or Bill talk* translates as (274), so that *sees* is forced to distribute over the disjunction:

(271) John sees Mary walk or Bill talk

(272) John sees Mary walk or John sees Bill talk

(273) $\text{SEE}(\hat{}\text{WALK}(m))(j) \vee \text{SEE}(\hat{}\text{TALK}(b))(j)$

(274) $\lambda w_{(s,t^1)}.[\check{}w(\hat{}\text{WALK}(m)) \vee \check{}w(\hat{}\text{TALK}(b))]$

6 Flexibility and Anaphora

In Section 2, some problematic aspects of using rules of quantification for the representation of quantifier scope ambiguities were discussed. In the present section we will go into problems concerning the second function of quantifying-in, the representation of anaphoric pronouns, and sketch how flexible interpretation can contribute to their solution.

First, the *syntax* of the rules (X),*i*, (XI),*i*, and (XII),*i* ('replace [...] by *appropriate anaphoric pronouns*') is hardly explicit. For example, it does not distinguish between pronouns and reflexives. Sentence (275) is incorrectly assigned the interpretation (276), an interpretation which had better be assigned to (277):

(275) Every man loves him

(276) $\forall v[\text{MAN}(v) \rightarrow \text{LOVE}(v)(v)]$

(277) Every man loves himself

Moreover, in its present format (replace 'the *first* occurrence of he_i by α, and the *remaining* occurrences by [...] pronouns'), the syntactic operation is unable to account for 'cataphoric pronouns' such as the word *its* in the present sentence. A successful attempt to make the syntax of quantifying-in less implicit has been made by Landman and Moerdijk (1983), who show that the syntactic binding conditions of Reinhart (1976) and Chomsky (1981) can be successfully incorporated into a Montagovian approach. The resulting theory is capable of handling the syntax of reflexives and cataphors without problems.

A second problem is that the *semantics* of quantifying-in amounts to a uniform analysis of anaphoric pronouns as 'bound variables', in the following sense: a syntactic variable he_i is translated into an **IL**-expression containing a free logical variable x_i. Furthermore, the formula which is the translation of the sentence[89] in which the syntactic variable occurs is turned into an expression which refers to a property by λ-abstraction over that logical variable. Noun phrases are translated into expressions referring to sets of properties, and the translation of the final sentence, with the noun phrase quantified in, is the result of applying the translation of the noun phrase to the intension of the λ-expression. The resulting formula asserts that the property expressed by the λ-expression belongs to the set of properties which is the denotation of the translation of the noun phrase.

However, this is problematic, as can be illustrated with anaphoric relationships between noun phrases in different sentences. Consider, for example, the first couple of lines of 'Frank's Wild Years', a song by Tom Waits:

> Frank settled down out in the valley. He hung his wild years on the nail that he drove through his wife's forehead. He sold used office furniture out down San Fernando Road, ...

If sentence concatenation is semantically interpreted as conjunction, and the rule scheme (X),i is applicable to sequences of sentences, then the interpretation of this particular song seems unproblematic. We start with the sequence *He$_8$ settled down in the valley. He$_8$ hung he$_8$'s wild years on the nail that he$_8$ drove through he$_8$'s wife's forehead. He$_8$ sold used office furniture out down San Fernando road, ...*, and then quantify in the noun phrase *Frank*.[90] But in general this approach makes wrong predictions. Suppose that *Frank* in the first line is replaced by *exactly one man*:

[89] We focus on rule (X),i.

[90] Note, however, that this result is obtained in a counterintuitive way: in order to interpret the song as a whole, we first have to generate the entire text with syntactic

> Exactly one man settled down out in the valley. He hung his wild years on the nail that he drove through his wife's forehead. He sold used office furniture out down San Fernando road, ...

If we were to derive this text by quantifying-in *exactly one man* in the the way indicated above, it will be interpreted as expressing that there is exactly one man who has the joint properties of settling down out in the valley, hanging his wild years on that nail, selling used office furniture, ... – which is wrong.[91] The modified text means that there is exactly one man who settled down out in the valley, and that *this* man hung up his wild years, etc. In both texts, the first sentence *introduces* an individual, and the second sentence gives a further description of that individual.

The inadequacy of quantifying-in can also be demonstrated on the basis of intrasentential anaphora. Not all anaphoric pronouns can be treated as bound variables, as various authors have argued (Geach (1962), Karttunen (1968a, 1968b, 1969), Partee (1970), Evans (1980), a.o.).

(278) A horse claims that it walks

(279) No horse claims that it walks

(280) Every farmer who owns a donkey beats it

(281) Every farmer who owns no donkey beats it

Both in (278) and (279), the pronoun *it* can be construed as being bound (in the sense of quantifying-in) by the subject of the matrix sentence. The sentences (280) and (281), on the other hand, display a contrast: *a donkey* in (280) can be an antecedent for *it*, whereas *no donkey* in (281) cannot. In such 'donkey sentences' (which owe their name to the particular examples in Geach (1962)), it appears to depend on the semantic nature of the possible antecedent whether an anaphoric link is permitted or not. However, both sentences can be obtained from *Every farmer who owns* he_3 *beats* he_3, by quantifying in *a donkey* and *no donkey*, respectively. Moreover, for sentence (280), this leads to interpretation (282) instead of the desired (283):

(282) $\exists v[\text{DONKEY}(v) \wedge \forall u[[\text{FARMER}(u) \wedge \text{OWN}(v)(u)] \rightarrow \text{BEAT}(v)(u)]]$

variables he_8 replacing the occurrences of *Frank*, *he*, *him* and *his*, way down to its last line *He*$_8$ *never could stand that dog*, before quantifying-in can take place.

[91] On this interpretation, the existence of other men who also settled down out in the valley but happened to be fond of that dog would be irrelevant for the truth of the text.

(283) $\forall u \forall v[[\text{FARMER}(u) \wedge \text{DONKEY}(v) \wedge \text{OWN}(v)(u)] \rightarrow \text{BEAT}(v)(u)]$

Kamp (1981) developed 'discourse representation theory' (DRT) in order to handle such phenomena. Similar proposals were made independently by Heim (1982) and Seuren (1985). DRT, which is not a grammar of isolated sentences but of texts (i.e., coherent sequences of sentences), is a formalization of Karttunen's (1968b) idea that indefinite noun phrases (and proper names) have what is commonly called 'context change potential', in that they introduce discourse referents in the domain of discourse, which can be denoted by subsequent anaphoric pronouns. By representing these discourse referents as so-called *reference markers*, DRT is able to obtain the correct interpretation for intersentential and 'donkey' anaphora. Moreover, DRT draws a sharp distinction between definite (*the printer*) and indefinite (*a printer*) noun phrases on the one hand, and quantified noun phrases (*every printer*, *no printer*) on the other, which enables it to account for the absence of 'donkey' and intersentential anaphora with the latter type of noun phrases, witness (281) and (284). (The symbol '*' in (284) indicates the impossibility of having an anaphoric link between the noun phrase it precedes and the word *it* in the second sentence.)

(284)
The printer A printer *Every printer *No printer	is in the next room. It has run out of paper.

DRT postulates a level of *discourse representation structure* which mediates between the levels of syntactic analysis and semantic interpretation. At this representational level, reference markers are introduced by indefinite noun phrases and proper names, and anaphoric links are established between pronouns and reference markers. This level, which constitutes the context of interpretation for the sequel of the text, has been claimed to be indispensable.

Nonetheless, Groenendijk and Stokhof (1991) show that, contrary to this claim, representationalism is not essential to the phenomena addressed by Kamp (1981), in the sense that replacing the traditional, static view on meaning by a procedural, dynamic one allows one to do without an intermediate level of representation. That is, these phenomena do not in themselves undermine the principle of compositionality, and the insights of DRT may well be incorporated into a compositional theory of model-theoretic semantics.

Thus, Groenendijk and Stokhof (1990) provide a 'dynamic Montague grammar' (DMG) for a fragment of English, in which English expressions (up to and including sequences of sentences) are compositionally translated into a language of dynamic intensional logic (**DIL**), a variant of Montague's **IL**.[92] **DIL** differs syntactically from **IL** in that it has an additional set of basic expressions of type e of so-called *discourse markers*. These discourse markers also figure in so-called *state switchers*: operators built up from a discourse marker d and an expression α of type e. State switchers can be put in front of an expression β of arbitrary type a, yielding an expression $\{\alpha/d\}\beta$ of the same type. The intensional parameters in **DIL** are not possible worlds, but so-called *states*. On account of two postulates, the *distinctness* and the *update postulate*, the set of states can be identified with the set of assignments of individuals to discourse markers.[93] Accordingly, the intension operator $^\wedge$ is interpreted as abstraction over assignments to discourse markers, and the extension operator $^\vee$ as application to the current assignment to discourse markers.

In the DMG fragment, (sequences of) sentences are translated into logical expressions of type $((s,t),t)$: the denotation of a sentence in a state s is a (characteristic function of a) set of propositions – viz., the propositions which are true after the sentence has been processed in s.[94] Of course, if a proposition p is in this set, then also any proposition weaker than p will be in it. So, a sentence φ is true – i.e., has some true possible continuation – in s iff the tautologous proposition (to be denoted by a special constant TRUE of type (s,t), or by $^\wedge[x = x]$) is a member of its denotation φ' in s.[95]

---

[92] **DIL** dates back to Janssen's (1986, Part 1, Chapter 4) semantics for assignment statements in programming languages.

[93] Discourse markers are interpreted as 'individual concepts', i.e., as functions from states to individuals, and their denotation in a state s is the result of applying their interpretation to s. The denotation of $\{\alpha/d\}\beta$ in a state s equals the denotation of β in the unique state s' which differs from s at most in that the denotation of d in s' equals the denotation of α in s. Putting a state switcher $\{\alpha/d\}$ in front of β has, roughly speaking, the semantic effect of performing a 'decent' substitution $[d{:=}\alpha]$ (i.e., no variables in α get bound) in β, with the exception (modulo λ-conversion and $^\vee{}^\wedge$-elimination) of subexpressions of β of the form $^\vee\gamma$ or $\{\alpha'/d'\}\gamma$.

[94] The meaning of (sequences of) sentences is an object of type $(s,((s,t),t))$: a function from states to sets of propositions. For each state s, this function yields the set of propositions which are true after the sentence has been processed in s.

[95] And φ's truth-conditional content is obtained from its truth value, $\varphi'(\text{TRUE})$, by abstraction over s. So, if $\varphi'_{((s,t),t)}$ is the dynamic denotation of a sentence φ, then φ's truth-conditional meaning of type (s,t) can be characterized as $^\wedge[\varphi'(\text{TRUE})]$. We see that the static notion of propositional sentence-meaning familiar from MG is retained in DMG,

Since the propositions which make up the extension of a sentence in a state s – the propositions which are true after the sentence has been processed in s – can be considered as jointly restricting the truth-conditions of its possible continuations in s, the following interpretation of sequencing (or concatenation) of sentences φ and ψ is in order: a proposition p will be true after the sequence of sentences $\varphi.\psi$ has been processed in state s if and only if the proposition that p holds after ψ has been processed is one of the propositions which are true after φ has been processed in s. Or, succinctly expressed in DIL, if and only if p is a member of the following set:

(285) $\lambda p.\varphi'(\hat{}\psi'(p))$

Because the intensional closure $\hat{}\psi'(p)$ of $\psi'(p)$ semantically amounts to an abstraction over assignments to discourse markers, sequencing of φ and ψ has the effect that occurrences of 'free' discourse markers d_i in ψ' can get bound by corresponding state switchers $\{\alpha/d_i\}$ in the function φ' which is applied to that intensional closure. By means of this, cross-sentential anaphora can be dealt with in a fully compositonal way. On the subsentential level, the same mechanism accounts for 'donkey' anaphora.

In the context of the present chapter it is worth mentioning that a grammar which handles both anaphora and scope ambiguities can be obtained by simply *adding* the type-shifting mechanism of Section 4 above to a 'best case' version of Groenendijk and Stokhof's DMG. Consider, for example, the text in (286). Assume the basic translations in (287) and let τ abbreviate the type $((s,t),t)$:[96]

(286) Every$_1$ woman finds a$_2$ man. He$_2$ walks.

---

but as a secondary, derived notion.

[96]The types of the variables, discourse markers and constants used in (287) are as follows: $P_{(s,((s,e),\tau))}$, $Q_{(s,((s,e),\tau))}$, $x_{(s,e)}$, $y_{(s,e)}$, $d_{1,e}$, $d_{2,e}$, $\text{WOMAN}_{(e,t)}$, $\text{FIND}_{(e,(e,t))}$, $\text{MAN}_{(e,t)}$, and $\text{WALK}_{(e,t)}$. (In contrast with the non-uniform treatment of noun phrases in DRT alluded to above, all noun phrases in DMG are uniformly assgned a translation of type $((s,((s,e),\tau)),\tau)$. In the presence of value raising, the indexed pronoun *he*$_2$ can also be assigned the translation d_2 of type e.)

The operators $\uparrow$, ;, $\mathcal{E}d_2$, $\Rightarrow$ and $\mathcal{A}d_1$ are all defined in terms of the familiar ones and of the state switcher. Let p and u be variables of respective types (s,t) and e. Then for φ_t, Φ_τ and Ψ_τ: $\uparrow\varphi = \lambda p.[\varphi \wedge \check{}p]$, $[\Phi\,;\Psi] = \lambda p.\Phi(\hat{}\Psi(p))$ (i.e., (285)); and $\mathcal{E}d_i\Phi = \lambda p.\exists u\{u/d_i\}[\Phi(p)]$ for $i \in \{1,2\}$. Finally, let $\sim\Phi$ equal $\lambda p.[\neg\Phi(\text{TRUE}) \wedge \check{}p]$. Then $[\Phi \Rightarrow \Psi] = \sim[\Phi\,;\sim\Psi]$ and $\mathcal{A}d_1\Phi = \sim\mathcal{E}d_1\sim\Phi$.

(287) $every_1$ $\lambda P\lambda Q.\mathcal{A}d_1[{}^{\vee}P({}^{\wedge}d_1) \Rightarrow {}^{\vee}Q({}^{\wedge}d_1)]$
woman $\lambda x.\uparrow\text{WOMAN}({}^{\vee}x)$
finds $\lambda x\lambda y.\uparrow\text{FIND}({}^{\vee}x)({}^{\vee}y)$
a_2 $\lambda P\lambda Q.\mathcal{E}d_2[{}^{\vee}P({}^{\wedge}d_2)\,;\,{}^{\vee}Q({}^{\wedge}d_2)]$
man $\lambda x.\uparrow\text{MAN}({}^{\vee}x)$
he_2 $\lambda P.{}^{\vee}P({}^{\wedge}d_2)$
walks $\lambda x.\uparrow\text{WALK}({}^{\vee}x)$

On account of argument raising, text (286) is assigned two interpretations of type τ. The lexical translation of *finds* is of type $((s,e),((s,e),\tau))$, and there are two non-equivalent ways to argument-raise it in two steps to type $((s,((s,((s,e),\tau)),\tau)),((s,((s,((s,e),\tau)),\tau)),\tau))$.[97] Interpretation (288) results if the first argument is raised first. The reverse order yields (289):[98]

(288) $\lambda X_1\lambda X_2.{}^{\vee}X_2({}^{\wedge}\lambda x_2.{}^{\vee}X_1({}^{\wedge}\lambda x_1.\uparrow\text{FIND}({}^{\vee}x_1)({}^{\vee}x_2)))$

(289) $\lambda X_1\lambda X_2.{}^{\vee}X_1({}^{\wedge}\lambda x_1.{}^{\vee}X_2({}^{\wedge}\lambda x_2.\uparrow\text{FIND}({}^{\vee}x_1)({}^{\vee}x_2)))$

The interpretations of the non-sentential expressions are combined by means of (intensional) functional application, and the interpretations of the two sentences are composed in the way indicated in (285). Using (288) and (289), respectively, the above text is assigned the following two reduced dynamic interpretations of type τ:

(290) $\lambda p.[[\forall v[\text{WOMAN}(v) \rightarrow \exists u[\text{MAN}(u) \wedge \text{FIND}(u)(v)]] \wedge \text{WALK}(d_2)] \wedge {}^{\vee}p]$

(291) $\lambda p.\exists u[[[\text{MAN}(u) \wedge \forall v[\text{WOMAN}(v) \rightarrow \text{FIND}(u)(v)]] \wedge \text{WALK}(u)] \wedge \{u/d_2\}{}^{\vee}p]$

By applying these to TRUE, we get the static interpretations of type t, which reduce to the following formulas:

(292) $[\forall v[\text{WOMAN}(v) \rightarrow \exists u[\text{MAN}(u) \wedge \text{FIND}(u)(v)]] \wedge \text{WALK}(d_2)]$

(293) $\exists u[[\text{MAN}(u) \wedge \forall v[\text{WOMAN}(v) \rightarrow \text{FIND}(u)(v)]] \wedge \text{WALK}(u)]$

[97] Cf. (160) and (161) above.
[98] X_1 and X_2 are variables of type $(s,((s,((s,e),\tau)),\tau))$.

It turns out that he_2 can be bound by *a man*, provided that this object noun phrase has wider scope than *every woman*.[99]

An interesting contribution in this respect is Dekker (1990; Chapter 2 and 3 of (1993)). Focusing on the notion of negation in a dynamic Montague grammar, Dekker argues that sentences should be assigned interpretations of type $((s,((s,t),t)),t)$ – the type of generalized quantifiers over propositions.[100] This claim is substantiated in the guise of a structural, conservative modification of DMG into the system DMG(2). Besides, Dekker shows that the DMG(2)-style dynamic interpretations of expressions can be obtained in a systematic way from simple static translations, by means of a version of the system of type change presented in Section 4 above. The resulting flexible dynamic Montague grammar is shown to be more than the sum of its flexible and dynamic parts: it applies successfully to known and new examples which exhibit puzzling and complex anaphoric dependencies.

[99]Moreover, the state switcher $\{u/d_2\}$ in (291) indicates that *a man* is still able to bind further pronouns on this reading. See Groenendijk and Stokhof (1990), Section 7, for discussion. See also Karttunen (1969a), pp. 6–7, a.o.

[100]Groenendijk and Stokhof (1990) propose a static negation (see the previous footnote: $\sim\Phi = \lambda p.[\neg\Phi(\text{TRUE})\wedge^{\vee}p]$; this negation is called 'static' since it disallows binding of further pronouns by noun phrase interpretations within Φ). However, they also observe that this negation cannot account for the possibility of anaphora in the text *It is not the case that John does not own* a_1 *car.* It_1 *is red and* it_1 *is parked in front of the house.* Therefore, they define an alternative, dynamic negation (viz., $\sim\Phi = \lambda p.\neg\Phi(p)$, in their discussion of possible extensions of the framework), but point out that it has unpleasant properties, for instance: $\sim\Phi;\Psi = \sim[\Phi;\Psi]$. Hence, *John*$_1$ *does not walk in the park.* He_1 *whistles* will be interpreted as expressing 'John does not both walk in the park and whistle'. Groenendijk and Stokhof also propose a possible solution, but it is rather ad hoc. Dekker shows that what one needs is a dynamic notion of negation which (*i*) obeys the law of double negation – including preservation of binding potential under double negation –; and (*ii*) involves negation of truth-conditional content (formally expressed: (*i*) $\sim\sim\Phi = \Phi$, and (*ii*) $\sim\uparrow\varphi = \uparrow\neg\varphi$). The static (dynamic) negation of DMG only satisfies the second (first) requirement. Dekker argues that it is impossible to obtain a notion of negation meeting both requirements if sentence denotations are of type $((s,t),t)$, expressing sets of propositions, but that at the slightly higher level of type $((s,((s,t),t)),t)$ – that of generalized quantifiers over propositions – it is possible to define an adequate notion of negation.

Appendix: Semantic Properties of the Fragment

The Fragment
Let $e^1 = ((s,e),t)$ and $e^2 = ((s,((s,e),t)),t)$. Then:

$$t_B(\mathrm{S}) = t \qquad t_B(\mathrm{TV}) = ((s,e),e^1)$$
$$t_B(\mathrm{T}) = e \qquad t_B(\mathrm{PV}) = ((s,t),e^1)$$
$$t_B(\mathrm{CN}) = t_B(\mathrm{IV}) = e^1 \qquad t_B(\mathrm{Det}) = ((s,e^1),e^2)$$

Let $C \in \{\mathrm{S,T,CN,IV,TV,PV,Det}\}$. The set $T(C)$ is the smallest set such that for all types b, d and finite sequences of types $\vec{a} = a_1, ..., a_n$; $\vec{c} = c_1, ..., c_m$:

$t_B(C) \in T(C)$,
$(\vec{a},((s,((s,b),d)),d)) \in T(C)$ if $(\vec{a},b) \in T(C)$, and
$(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d))) \in T(C)$ iff $(\vec{a},((s,b),(\vec{c},d))) \in T(C)$

Next, let $\vec{x} = v_{1,a_1}, \ldots, v_{n,a_n}$; $\vec{y} = v_{(n+1)+1,c_1}, \ldots, v_{(n+1)+m,c_m}$; $z = v_{n+1,(s,b)}$; $u = v_{n+1,(s,((s,b),d))}$; $w = v_{n+1,(s,((s,((s,b),d)),d))}$; let $\mathrm{FA}(\alpha',\beta') = \alpha'(\hat{}\beta')$ for α' of type $((s,a),b)$ and β' of type a; and let $\mathrm{GC}(\alpha',\beta') = \lambda\vec{x}_{\vec{a}}.[\alpha'(\vec{x}) \wedge \beta'(\vec{x})]$ and $\mathrm{GD}(\alpha',\beta') = \lambda\vec{x}_{\vec{a}}.[\alpha'(\vec{x}) \vee \beta'(\vec{x})]$ for α' and β' of type $(\vec{a},t)$.

For every expression γ, the translation set $Tr(\gamma)$ of γ is the smallest set such that $Tr_B(\gamma) \subseteq Tr(\gamma)$ and for γ', γ'' and γ''' of type $(\vec{a},b)$, $(\vec{a},((s,b),(\vec{c},d)))$ and $(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d)))$, respectively:

[VR] if $\gamma' \in Tr(\gamma)$, then $\lambda\vec{x}\lambda u.^{\vee}u(\hat{}[\gamma'(\vec{x})]) \in Tr(\gamma)$,
[AR] if $\gamma'' \in Tr(\gamma)$, then $\lambda\vec{x}\lambda w\lambda\vec{y}.^{\vee}w(\hat{}\lambda z.\gamma''(\vec{x})(z)(\vec{y})) \in Tr(\gamma)$,
[AL] if $\gamma''' \in Tr(\gamma)$, then $\lambda\vec{x}\lambda z\lambda\vec{y}.\gamma'''(\vec{x})(\hat{}\lambda u.^{\vee}u(z))(\vec{y}) \in Tr(\gamma)$;

where $Tr_B(\gamma) = \{tr_B(\gamma)\}$ for lexical expressions γ and $tr_B(\gamma)$ listed in (117); $Tr_B(\gamma) = \{\mathrm{FA}(\alpha',\beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$ if γ is produced by rule

(I) $[_{\mathrm{T}}\alpha] + [_{\mathrm{IV}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{S}}[_{\mathrm{T}}\alpha][_{\mathrm{IV}}\beta]]$
(II) $[_{\mathrm{TV}}\alpha] + [_{\mathrm{T}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{IV}}[_{\mathrm{TV}}\alpha][_{\mathrm{T}}\beta]]$,
(III) $[_{\mathrm{Det}}\alpha] + [_{\mathrm{CN}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{T}}[_{\mathrm{Det}}\alpha][_{\mathrm{CN}}\beta]]$, or
(IV) $[_{\mathrm{PV}}\alpha] + [_{\mathrm{S}}\beta] \rightarrow \gamma$, where $\gamma = [_{\mathrm{IV}}[_{\mathrm{PV}}\alpha][_{\mathrm{S}}\beta]]$;

$Tr_B(\gamma) = \{\mathrm{GC}(\alpha',\beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$ if γ is produced by a rule $[_C\alpha] + [_C\beta] \rightarrow \gamma$, where $\gamma = [_C[_C\alpha]$ *and* $[_C\beta]]$ and $C \in \{\mathrm{S,IV,TV,CN}\}$; and $Tr_B(\gamma) = \{\mathrm{GD}(\alpha',\beta') \mid \alpha' \in Tr(\alpha) \text{ and } \beta' \in Tr(\beta)\}$ if γ is produced by a rule $[_C\alpha]+[_C\beta] \rightarrow \gamma$, where $\gamma = [_C[_C\alpha]$ *or* $[_C\beta]]$ and $C \in \{\mathrm{S,IV,T,TV,CN}\}$.

Notation Conventions
We will write $a \Rightarrow b$ ($\alpha_a \Rightarrow \beta_b$) iff a number of applications of VR, AR and AL to (a term α of) type a yield (a term β of) type b. We will abbreviate types of the form $((s,a),b)$ as $(a;b)$. Note that, given this notation, the operations VR, AR and AL give rise to the following one-step type transitions (i.e, when applied to a term of the left-hand side type, they yield a term of the right-hand side type):

(i) $(\vec{a},b) \Rightarrow (\vec{a},((b;d);d))$,
(ii) $(\vec{a},(b;(\vec{c},d))) \Rightarrow (\vec{a},(((b;d);d);(\vec{c},d)))$, and
(iii) $(\vec{a},(((b;d);d);(\vec{c},d))) \Rightarrow (\vec{a},(b;(\vec{c},d)))$.

An application of VR (AR, AL) which leads to the type transition (i) ((ii), (iii)) will be denoted by VR_i (AR_i, AL_i) iff $\mathrm{length}(\vec{a}) = i-1$. The type d introduced by an application of VR_i (AR_i) (cf. (i) ((ii))) is called its *exponent*. An application of VR_i (AR_i) with exponent d will be denoted by $\mathrm{VR}_{i,d}$ ($\mathrm{AR}_{i,d}$). For example:

$$((s,e),t) = (e;t) \Rightarrow_{\mathrm{VR}_{1,t}} (((e;t);t);t) = ((s,((s,((s,e),t)),t)),t)$$
$$((s,e),t) = (e;t) \Rightarrow_{\mathrm{VR}_{2,t}} (e;((t;t);t)) = ((s,e),((s,((s,t),t)),t))$$
$$((s,e),t) = (e;t) \Rightarrow_{\mathrm{VR}_{1,e}} (((e;t);e);e) = ((s,((s,((s,e),t)),e)),e)$$
$$((s,e),t) = (e;t) \Rightarrow_{\mathrm{VR}_{2,e}} (e;((t;e);e)) = ((s,e),((s,((s,t),e)),e))$$

Value Raising $\cap$ *Argument Raising*
A $\mathrm{VR}_{i,d}$ type transition $(\vec{a},b) \Rightarrow_{\mathrm{VR}_{i,d}} (\vec{a},((b;d);d))$ where $b = (c;d)$ can also be achieved via $\mathrm{AR}_{i,d}$: $(\vec{a},(c;d)) \Rightarrow_{\mathrm{AR}_{i,d}} (\vec{a},(((c;d);d);d))$. Observe that the translations thus derived are equivalent by η-conversion: $\lambda\vec{x}\lambda w.^{\vee}w(^{\wedge}[\alpha(\vec{x})]) \Leftrightarrow \lambda\vec{x}\lambda w.^{\vee}w(^{\wedge}\lambda z.\alpha(\vec{x})(z))$.[1] In view of this overlap, we will denote applications of VR where $b \neq (c;d)$ by NR (Non-Argument Raising) and applications of VR where $b = (c;d)$ by VAR (Value-cum-Argument Raising).

Argument Lowering
It can be shown that in the derivation of a translation applications of AL can be ordered before applications of AR; that applications of AL_i can be ordered before applications of VR_j if $i \leq j$; that applications of AL can be

[1] Throughout this Appendix we will exploit (a) that (both basic and derived) translations of expressions do not contain free variables; and (b) that the variables introduced by applications of VR, AR, AL, GC and GD are all different.

ordered before applications of FA, GC and GD; and that the relative order of different AL-applications to a translation is semantically irrelevant:

(1) (*a*) $\mathrm{AL}_i(\mathrm{VR}_i(\alpha)) \Leftrightarrow \alpha$,
(*b*) $\mathrm{AL}_i(\mathrm{AR}_i(\alpha)) \Leftrightarrow \alpha$,
(*c*) If $i \neq j$, then $\mathrm{AL}_i(\mathrm{AR}_j(\alpha)) \Leftrightarrow \mathrm{AR}_j(\mathrm{AL}_i(\alpha))$,
(*d*) If $i < j$, then $\mathrm{AL}_i(\mathrm{VR}_j(\alpha)) \Leftrightarrow \mathrm{VR}_j(\mathrm{AL}_i(\alpha))$.

(*a*) If AL_i is applied immediately after VR_i, then the type of $\mathrm{VR}_i(\alpha)$ is $(\vec{a}, (((c; d); d); d))$, where $\mathrm{length}(\vec{a}) = i - 1$. Hence VR_i is an instance of VAR, and the reader is referred to (*b*).
(*b*) If AL_i is applied immediately after AR_i, then the type of $\mathrm{AR}_i(\alpha)$ is $(\vec{a}, (((b; d); d); (\vec{c}, d)))$ and $\mathrm{AL}_i(\mathrm{AR}_i(\alpha))$ and α are both of type $(\vec{a}, (b; (\vec{c}, d)))$. Their equivalence can be shown by λ-conversion and $^\vee\!\wedge$-elimination: $\alpha \Leftrightarrow \lambda\vec{x}\lambda z\lambda\vec{y}.[\,\lambda\vec{x}\lambda w\lambda\vec{y}.{}^\vee w(\wedge\lambda z.\alpha(\vec{x})(z)(\vec{y}))\,](\vec{x})(\wedge\lambda u.{}^\vee u(z))(\vec{y}) = \mathrm{AL}_i(\mathrm{AR}_i(\alpha))$.
(*c*) If AL_i is applied immediately after AR_j and $i < j$, then the type of $\mathrm{AR}_j(\alpha) = (\vec{a}, (((b; b'); b'); (\vec{c}, (((d; d'); d'); f))))$, $\mathrm{length}(\vec{a}) = i - 1$, $\mathrm{length}(\vec{c}) = (j-1) - i$ and α and $\mathrm{AL}_i(\mathrm{AR}_j(\alpha))$ have type $(\vec{a}, (b; (\vec{c}, (((d; d'); d'); f))))$. If AL_i is applied immediately after AR_j and $i > j$, then the type of $\mathrm{AR}_j(\alpha) = (\vec{a}, (((b; b'); b'); (\vec{c}, (((d; d'); d'); f))))$, $\mathrm{length}(\vec{a}) = j - 1$ and $\mathrm{length}(\vec{c}) = (i-1)-j$, while α and $\mathrm{AL}_i(\mathrm{AR}_j(\alpha))$ have type $(\vec{a}, (((b; b'); b'); (\vec{c}, (d; f))))$. In both cases, the equivalence of α and $\mathrm{AL}_i(\mathrm{AR}_j(\alpha))$ is shown by λ-conversion and $^\vee\!\wedge$-elimination. The last case, (*d*), where AL_i applies immediately after AR_j and $i < j$, is analogous to the subcase $i < j$ of (*c*).

(2) (*a*) $\mathrm{AL}_i(\mathrm{FA}(\alpha, \beta)) = \mathrm{FA}(\mathrm{AL}_{i+1}(\alpha), \beta)$
(*b*) $\mathrm{AL}_i(\mathrm{GC}(\alpha, \beta)) = \mathrm{GC}(\mathrm{AL}_i(\alpha), \mathrm{AL}_i(\beta))$
(*c*) $\mathrm{AL}_i(\mathrm{GD}(\alpha, \beta)) = \mathrm{GD}(\mathrm{AL}_i(\alpha), \mathrm{AL}_i(\beta))$

Proofs by λ-conversion and $^\vee\!\wedge$-elimination. For instance, $\mathrm{AL}_i(\mathrm{GC}(\alpha, \beta)) = \lambda\vec{x}\lambda z\lambda\vec{y}.[\lambda\vec{x}\lambda w\lambda\vec{y}.[\alpha(\vec{x})(w)(\vec{y}) \wedge \beta(\vec{x})(w)(\vec{y})]](\vec{x})(\wedge\lambda u.{}^\vee u(z))(\vec{y})$ as well as $\mathrm{GC}(\mathrm{AL}_i(\alpha), \mathrm{AL}_i(\beta)) =$

$$\lambda\vec{x}\lambda w\lambda\vec{y}.[[\lambda\vec{x}\lambda z\lambda\vec{y}.\alpha(\vec{x})(\wedge\lambda u.{}^\vee u(z))(\vec{y})](\vec{x})(w)(\vec{y}) \wedge [\lambda\vec{x}\lambda z\lambda\vec{y}.\beta(\vec{x})(\wedge\lambda u.{}^\vee u(z))(\vec{y})](\vec{x})(w)(\vec{y})]$$

reduce to $\lambda\vec{x}\lambda z\lambda\vec{y}.[\alpha(\vec{x})(\wedge\lambda u.{}^\vee u(z))(\vec{y}) \wedge \beta(\vec{x})(\wedge\lambda u.{}^\vee u(w))(\vec{y})]$.

(3) ${}^i\mathrm{AL}({}^j\mathrm{AL}(\alpha)) \Leftrightarrow {}^j\mathrm{AL}({}^i\mathrm{AL}(\alpha))$.

Proof by λ-conversion and $^\vee\!\wedge$-elimination.

As a consequence of (1) through (3) and the reduction to minimal exponents[2] to be discussed below, AL can be assumed to apply only to lexical tranlations and before all other type shifts.[3]

The following counterparts of (2)(*a*) hold for applications of AR and VR:

(4) (*a*) $\mathrm{AR}_{i,d}(\mathrm{FA}(\alpha,\beta)) \Leftrightarrow \mathrm{FA}(\mathrm{AR}_{i+1,d}(\alpha),\beta)$
(*b*) $\mathrm{VR}_{i,d}(\mathrm{FA}(\alpha,\beta)) \Leftrightarrow \mathrm{FA}(\mathrm{VR}_{i+1,d}(\alpha),\beta)$

Proof of (*a*): $\mathrm{AR}_{i,d}(\mathrm{FA}(\alpha,\beta)) = \lambda\vec{x}\lambda w\lambda\vec{y}.^{\vee}w(^{\wedge}\lambda z.\alpha(^{\wedge}\beta)(\vec{x})(z)(\vec{y}))$. Note that $\mathrm{AR}_{i+1,d}(\alpha) = \lambda v\lambda\vec{x}\lambda w\lambda\vec{y}.^{\vee}w(^{\wedge}\lambda z.\alpha(v)(\vec{x})(z)(\vec{y}))$, and that therefore $\mathrm{FA}(\mathrm{AR}_{i+1,d}(\alpha),\beta) = \mathrm{AR}_{i+1,d}(\alpha)(^{\wedge}\beta) \Leftrightarrow \lambda\vec{x}\lambda w\lambda\vec{y}.^{\vee}w(^{\wedge}\lambda z.\alpha(^{\wedge}\beta)(\vec{x})(z)\vec{y}))$ via λ-conversion and $^{\vee}{}^{\wedge}$-elimination. The proof of (*b*) is analogous. Note that there are no counterparts of (2)(*b*) and (*c*) for VR and AR. Given that AL can be assumed to apply to lexical translations only, it follows from (4) that all type change can be assumed to apply to lexical items, conjunctions and disjunctions.

Eliminating Non-Argument Raising

Let $\mathrm{length}(\vec{a}) = i-1$ and $\mathrm{length}(\vec{c}) = j-1$. We will now consider derived translations φ' and α', where φ' is the result of applying NR_i to φ, and where α' of type $(\vec{c}', ((b;c);d))$ has been derived from α of type $(\vec{c}, b)$ without AL_j and possibly using VR_k, AR_k and AL_k for $k > j$, but only AR^{μ}_k (i.e., 'minimal' AR_k with an atomic exponent) if $k < j$. We show that if α' is the main functor of the i-th argument of φ', then the application of NR_i can be eliminated, so to speak, since there are translations φ'' (derived from φ without NR_i) and α'' (derived from α) such that for all sequences $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}_{\vec{c}}$ of different variables it holds that $\varphi''(\vec{\psi})(^{\wedge}[\alpha''(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})(^{\wedge}[\alpha'(\vec{\chi})])$:[4]

(5) If $\varphi_{(\vec{a},(b;c))} \Rightarrow_{\mathrm{NR}_{i,d}} \varphi'_{(\vec{a},(((b;c);d);d))}$ and $\alpha_{(\vec{c},b)} \Rightarrow_{\$} \alpha'_{(\vec{c}',((b;c);d))}$, where \$ contains no AL_j and, if $k < j$, only AR^{μ}_k, then there are $\varphi_{(\vec{a},(b;c))} \Rightarrow_{\$\$} \varphi''_{(\vec{a},(b';d))}$ and $\alpha_{(\vec{c},b)} \Rightarrow_{\$\$\$} \alpha''_{(\vec{c}',b')}$ such that
(*a*) \$\$ only contains VAR_i and VR_k, AR_k, AL_k for $k > i$;
(*b*) \$\$\$ only contains VR_j and AR^{μ}_k for $k < j$; and
(*c*) for all $\vec{\psi}_{\vec{a}}$, $\vec{\chi}_{\vec{c}'}$: $\varphi''(\vec{\psi})(^{\wedge}[\alpha''(\vec{\chi})]) \Leftrightarrow \alpha'(\vec{\chi})(^{\wedge}[\varphi(\vec{\psi})])$.

[2] If an application $(\vec{a}, b) \Rightarrow (\vec{a}, ((b;d);d))$ of VR has an atomic exponent type d, then it introduces a lowerable argument only if $b = (c;d)$, i.e.: only if it is an application of VAR.

[3] Since AL leads to less complex types, we could eliminate it and replace it by (finitely many) additional lexical translations – in the present fragment $\lambda x.\textsc{be-missing}(^{\wedge}\lambda P.^{\vee}P(x))$ and $\lambda y\lambda x.\textsc{seek}(^{\wedge}\lambda P.^{\vee}P(y))(^{\vee}x)$ for *is missing* and *seeks*, respectively.

[4] Note that we have $\varphi'(\vec{\psi})(^{\wedge}[\alpha'(\vec{\chi})]) \Leftrightarrow \alpha'(\vec{\chi})(^{\wedge}[\varphi(\vec{\psi})])$ since $\varphi' = \lambda\vec{x}\lambda u.^{\vee}u(^{\wedge}[\varphi(\vec{x})])$.

Proof by induction on q, the number of AR_k^μ for $k < j$ in \$:
• $q = 0$. Then α' is of type $(\vec{c}, ((b;c);d))$ and \$ consists of $\mathrm{V(A)R}_{j,c}$ followed by VR_k, AR_k, AL_k for $k > j$, so α' reduces to a term $\lambda\vec{y}\lambda u.\mathrm{A}\langle {}^\vee u({}^\wedge[\alpha(\vec{y})])\rangle$, where $\vec{y}$ and u do not occur in A.[5] Now, for \$\$ = $\mathrm{VAR}_{i,c}$ followed by \$* (i.e., \$ with $\mathrm{VR}_{j+m,d}$, $\mathrm{AR}_{j+m,d}$, AL_{j+m} replaced by $\mathrm{VR}_{i+m,d}$, $\mathrm{AR}_{i+m,d}$, AL_{i+m}, respectively) and \$\$\$ = $\mathrm{VR}_{j,c}$, we have:

$\varphi_{(\vec{a},(b;c))} \Rightarrow_{\mathrm{VAR}_i} \lambda\vec{x}\lambda u'.{}^\vee u'({}^\wedge[\varphi(\vec{x})])_{(\vec{a},(((b;c);c);c))} \Rightarrow_{\$^*} \varphi''_{(\vec{a},((((b;c);c);d))}$ and
$\alpha_{(\vec{c},b)} \Rightarrow_{\mathrm{VR}_j} \lambda\vec{y}\lambda u.{}^\vee u({}^\wedge[\alpha(\vec{y})])_{(\vec{c},((b;c);c))}$.

Note that φ'' reduces to $\lambda\vec{x}\lambda u'.\mathrm{A}\langle {}^\vee u'({}^\wedge[\varphi(\vec{x})])\rangle$, where $\vec{x}$ and u' do not occur in A, so that for $\alpha'' = \lambda\vec{y}\lambda u.{}^\vee u({}^\wedge[\alpha(\vec{y})])$ it holds that $\varphi''(\vec{\psi})({}^\wedge[\alpha''(\vec{\chi})]) \Leftrightarrow \mathrm{A}\langle\varphi(\vec{\psi})({}^\wedge[\alpha(\vec{\chi})])\rangle \Leftrightarrow \alpha'(\vec{\chi})({}^\wedge[\varphi(\vec{\psi})])$.
• $q > 0$. Then \$ is as follows (where AR_k^μ is the q-th (last) AR_k^μ for $k < j$):

$\alpha_{(\vec{c},b)} \Rightarrow_{\$^1} \alpha^\circ_{(\vec{c}',(c'';(\vec{c}''',((b;c);d'))))} \Rightarrow_{\mathrm{AR}_k^\mu}$
$\lambda\vec{y}'\lambda y''\lambda\vec{y}'''\lambda v\lambda\vec{w}.{}^\vee y''({}^\wedge\lambda z.\alpha^\circ(\vec{y}')(z)(\vec{y}''')(v)(\vec{w}))_{(\vec{c}',(((c'';f);f);(\vec{c}''',((b;c);d))))}$
$\Rightarrow_{\$^2} \alpha'_{(\vec{c}',(((c'';f);f);(\vec{c}''',((b;c);d))))}$

Derivation $\$^2$ contains only VR_k, AR_k and AL_k for $k > j$, so α' reduces to $\lambda\vec{y}'\lambda y''\lambda\vec{y}'''\lambda v.\mathrm{A}\langle {}^\vee y''({}^\wedge\lambda z.\alpha^\circ(\vec{y}')(z)(\vec{y}''')(v)(\vec{w}))\rangle$, where $\vec{y}', y'', \vec{y}''', v$ do not occur in A. Observe that for $\vec{\psi}_{\vec{a}}, \vec{\chi}'_{\vec{c}'}, \chi''_{(s,((c'';f);f))}, \vec{\chi}'''_{\vec{c}'''}$ and $\varphi_{(\vec{a},(b;c))}$ we have $\alpha'(\vec{\chi}')(\chi'')(\vec{\chi}''')({}^\wedge[\varphi(\vec{\psi})]) \Leftrightarrow \mathrm{A}\langle {}^\vee\chi''({}^\wedge\lambda z.\alpha^\circ(\vec{\chi}')(z)(\vec{\chi}''')({}^\wedge[\varphi(\vec{\psi})])(\vec{w}))\rangle$. Since derivation $\$^1$ contains $q-1$ times AR_k^μ for $k < j$, by induction hypothesis there are $\varphi_{(\vec{a},(b;c))} \Rightarrow_{\$\$^*} \varphi^*_{(\vec{a},(b';d'))}$ and $\alpha_{(\vec{c},b)} \Rightarrow_{\$\$\$^*} \alpha^*_{(\vec{c}',(c'';(\vec{c}''',b')))}$, where \$\$* contains only VAR_i and VR_k, AR_k, AL_k for $k > i$; \$\$\$* contains only VR_j and AR_k^μ for $k < j$; and for all $\vec{\psi}, \vec{\chi}', z_{(s,c')}, \vec{\chi}'''$: $\varphi^*(\vec{\psi})({}^\wedge\alpha^*(\vec{\chi}')(z)(\vec{\chi}''')) \Leftrightarrow \alpha^\circ(\vec{\chi}')(z)(\vec{\chi}''')({}^\wedge[\varphi(\vec{\psi})])$. Now, for \$\$ = \$\$* followed by VAR_i and $\$^{2*}$ (i.e.: $\$^2$ with $\mathrm{VR}_{j+m,d}$, $\mathrm{AR}_{j+m,d}$, AL_{j+m} replaced by $\mathrm{VR}_{i+m,d}$, $\mathrm{AR}_{i+m,d}$, AL_{i+m}, respectively) and \$\$\$ = \$\$\$* followed by VR_i and AR_k^μ, we have:

$\varphi_{(\vec{a},(b;c))} \Rightarrow_{\$\$^*} \varphi^*_{(\vec{a},(b';d'))} \Rightarrow_{\mathrm{VAR}_i} \lambda\vec{x}\lambda u.{}^\vee u({}^\wedge[\varphi^*(\vec{x})])_{(\vec{a},(((b';d');d');d'))}$
$\Rightarrow_{\$^{2*}} \varphi''_{(\vec{a},(((b';d');d');d))}$ and
$\alpha_{(\vec{c},b)} \Rightarrow_{\$\$\$^*} \alpha^*_{(\vec{c}',(c'';(\vec{c}''',b')))}$
$\Rightarrow_{\mathrm{VR}_i} \lambda\vec{y}'\lambda z\lambda\vec{y}'''\lambda v.{}^\vee v({}^\wedge[\alpha^*(\vec{y}')(z)(\vec{y}''')])_{(\vec{c}',(c'';(\vec{c}''',((b';d');d'))))}$
$\Rightarrow_{\mathrm{AR}_k^\mu} \alpha''_{(\vec{c}',(((c'';f);f);(\vec{c}''',((b';d');d'))))}$.

[5] We will use symbols $\mathrm{A}\langle\,\rangle$, $\Phi\langle\,\rangle$, ... for *contexts*. 'A context $\mathrm{A}\langle\,\rangle$ is a term with some holes in it. If $\mathrm{A}\langle\,\rangle$ is a context and α is a term, then $\mathrm{A}\langle\alpha\rangle$ denotes the result of placing α in the holes of $\mathrm{A}\langle\,\rangle$. In this act free variables of α may become bound in $\mathrm{A}\langle\alpha\rangle$.' (Barendregt (1981), p. 29). We will only encounter contexts with one hole.

Note that φ'' reduces to $\lambda\vec{x}\lambda u.\mathrm{A}\langle{}^{\vee}u({}^{\wedge}[\varphi^*(\vec{x})])(\vec{w})\rangle$, where $\vec{x}$, u do not occur in A, while α'' reduces to $\lambda\vec{y}'\lambda y''\lambda\vec{y}'''\lambda v\lambda\vec{w}.{}^{\vee}y''({}^{\wedge}\lambda z.{}^{\vee}v({}^{\wedge}[\alpha^*(\vec{y}')(z)(\vec{y}''')])(\vec{w}))$. So $\varphi''(\vec{\psi})({}^{\wedge}\alpha''(\vec{\chi}')(\chi'')(\vec{\chi}''')) \Leftrightarrow \mathrm{A}\langle{}^{\vee}\chi''({}^{\wedge}\lambda z.\varphi^*(\vec{\psi})({}^{\wedge}[\alpha^*(\vec{\chi}')(z)(\vec{\chi}''')])(\vec{w}))\rangle$, which is equivalent to $\mathrm{A}\langle{}^{\vee}\chi''({}^{\wedge}\lambda z.\alpha^\circ(\vec{\chi}')(z)(\vec{\chi}''')({}^{\wedge}[\varphi(\vec{\psi})])(\vec{w}))\rangle$ by induction hypothesis. We saw that this is equivalent to $\alpha'(\vec{\chi}')(\chi'')(\vec{\chi}''')({}^{\wedge}[\varphi(\vec{\psi})])$. □

Minimalizing Exponents

Again, let $\mathrm{length}(\vec{a}) = i - 1$ and $\mathrm{length}(\vec{c}) = j - 1$. Let d be an atomic type. This time, we will consider derived translations φ' and α' where φ' is the result of applying non-minimal $\mathrm{AR}_{i,(\vec{a}'',d)}$ to φ and where α' of type $(\vec{c}', ((b;(\vec{a}'',d));(\vec{a}'',d)))$ has been derived from α of type $(\vec{c}, b)$ by using $\mathrm{VR}_{j,(\vec{a}'',d)}$, possibly followed by AR^{μ}_{k} for $k < j$. Now, if $\mathrm{AR}_{i,(\vec{a}'',d)}$ and $\mathrm{VR}_{j,(\vec{a}'',d)}$ are replaced by $\mathrm{AR}_{j,d}$ and $\mathrm{VR}_{j,d}$, then for the resulting φ'' and α'' it holds that $\varphi''(\vec{\psi})({}^{\wedge}[\alpha''(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})({}^{\wedge}[\alpha'(\vec{\chi})])$ for all $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}_{\vec{c}}$:

(6) Let $\varphi_{(\vec{a},(b;(\vec{a}',(\vec{a}'',d))))} \Rightarrow_{\mathrm{AR}_{i,(\vec{a}'',d)}} \varphi'_{(\vec{a},(((b;(\vec{a}'',d));(\vec{a}'',d));(\vec{a}',(\vec{a}'',d))))}$ and $\alpha_{(\vec{c},b)} \Rightarrow_{\mathrm{VR}_{j,(\vec{a}'',d)}} \lambda\vec{y}\lambda v.{}^{\vee}v({}^{\wedge}[\alpha(\vec{y})])_{(\vec{c},((b;(\vec{a}'',d));(\vec{a}'',d)))} \Rightarrow_{\$} \alpha'_{(\vec{c}',((b;(\vec{a}'',d));(\vec{a}'',d)))}$, where \$ only contains AR^{μ}_{k} for $k < j$. Then for $\varphi_{(\vec{a},(b;(\vec{a}',(\vec{a}'',d))))} \Rightarrow_{\mathrm{AR}_{i,d}} \varphi''_{(\vec{a},(((b;d);d);(\vec{a}',(\vec{a}'',d))))}$ and $\alpha_{(\vec{c},b)} \Rightarrow_{\mathrm{VR}_{j,d}} \lambda\vec{y}\lambda v'.{}^{\vee}v'({}^{\wedge}[\alpha(\vec{y})])_{(\vec{c},((b;d);d))} \Rightarrow_{\$} \alpha''_{(\vec{c}',((b;d);d))}$ it holds that for all $\vec{\psi}_{\vec{a}}$, $\vec{\chi}_{\vec{c}'}$: $\varphi''(\vec{\psi})({}^{\wedge}[\alpha''(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})({}^{\wedge}[\alpha'(\vec{\chi})])$.

Proof: note that φ'' reduces to $\lambda\vec{x}\lambda w\lambda\vec{x}'\lambda\vec{x}''.{}^{\vee}w({}^{\wedge}\lambda z.\varphi(\vec{x})(z)(\vec{x}')(\vec{x}''))$ and that α'' reduces to $\lambda\vec{y}'\lambda v.\mathrm{A}\langle{}^{\vee}v({}^{\wedge}[\alpha(\vec{y})])\rangle$, where v does not occur in A. Moreover, φ' reduces to $\lambda\vec{x}\lambda w\lambda\vec{x}'\lambda\vec{x}''.{}^{\vee}w({}^{\wedge}\lambda z.\varphi(\vec{x})(z)(\vec{x}'))(\vec{x}'')$ and α' reduces to $\lambda\vec{y}'\lambda v'\lambda\vec{y}''.\mathrm{A}\langle{}^{\vee}v'({}^{\wedge}[\alpha(\vec{y})])(\vec{y}'')\rangle$, where v' and $\vec{y}''$ do not occur in A. Therefore, both $\varphi''(\vec{\psi})({}^{\wedge}[\alpha''(\vec{\chi})])$ and $\varphi'(\vec{\psi})({}^{\wedge}[\alpha'(\vec{\chi})])$ are equivalent to the term $\lambda\vec{x}'\lambda\vec{x}''.[\mathrm{A}\langle\varphi(\vec{\psi})({}^{\wedge}[\alpha(\vec{y})])(\vec{x}')(\vec{x}'')\rangle[\vec{y}' := \vec{\chi}]]$. □

Final Value Raising

Note that if derivation \$ in (6) is empty, then we have that for all $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}_{\vec{c}}$: $\varphi'(\vec{\psi})({}^{\wedge}[\alpha'(\vec{\chi})]) \Leftrightarrow \varphi(\vec{\psi})({}^{\wedge}[\alpha(\vec{\chi})])$:

(7) Let $\varphi_{(\vec{a},(b;(\vec{a}',(\vec{a}'',d))))} \Rightarrow_{\mathrm{AR}_{i,(\vec{a}'',d)}} \varphi'_{(\vec{a},(((b;(\vec{a}'',d));(\vec{a}'',d));(\vec{a}',(\vec{a}'',d))))}$ and $\alpha_{(\vec{c},b)} \Rightarrow_{\mathrm{VR}_{j,(\vec{a}'',d)}} \lambda\vec{y}\lambda v.{}^{\vee}v({}^{\wedge}[\alpha(\vec{y})])_{(\vec{c},((b;(\vec{a}'',d));(\vec{a}'',d)))} = \alpha'$. Then for all $\vec{\psi}_{\vec{a}}$, $\vec{\chi}_{\vec{c}'}$: $\varphi'(\vec{\psi})({}^{\wedge}[\alpha'(\vec{\chi})]) \Leftrightarrow \varphi(\vec{\psi})({}^{\wedge}[\alpha(\vec{\chi})])$.

Since φ' and α reduce to the terms $\lambda\vec{x}\lambda w\lambda\vec{x}'\lambda\vec{x}''.{}^{\vee}w({}^{\wedge}\lambda z.\varphi(\vec{x})(z)(\vec{x}'))(\vec{x}'')$ and $\lambda\vec{y}\lambda v.{}^{\vee}v({}^{\wedge}[\alpha(\vec{y})])$, respectively, the term $\varphi'(\vec{\psi})({}^{\wedge}[\alpha'(\vec{\chi})])$ is equivalent to $\lambda\vec{x}'\lambda\vec{x}''.\varphi(\vec{\psi})({}^{\wedge}[\alpha(\vec{\chi})])(\vec{x}')(\vec{x}'')$, and hence to $\varphi(\vec{\psi})({}^{\wedge}[\alpha(\vec{\chi})])$. □

Inheritance of Equivalence
The facts (5) through (7) above concern special circumstances in which terms $\varphi_{(\vec{a},(b;d))}$, $\varphi'_{(\vec{a},(b';d))}$, $\alpha_{(\vec{c},b)}$, and $\alpha'_{(\vec{c},b')}$ have the property that for all $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}_{\vec{c}}$: $\varphi(\vec{\psi})(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$. In (8) through (11) below we will consider situations in which such equivalences are inherited in a larger context.

(8) Let \$ consist of AR^{μ}_{k} for $k < i$ and $\mathrm{VR}_k, \mathrm{AR}_k, \mathrm{AL}_k$ for $k > i$, and let $\varphi \Rightarrow_{\$} \varphi''_{(\vec{a}',(b;d'))}$ and $\varphi' \Rightarrow_{\$} \varphi'''_{(\vec{a}',(b';d'))}$. Then for all $\vec{\psi}'_{\vec{a}'}$ and $\vec{\chi}_{\vec{c}}$: $\varphi''(\vec{\psi}')(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \varphi'''(\vec{\psi}')(\hat{}[\alpha'(\vec{\chi})])$.

Proof: observe that φ'' reduces to $\lambda\vec{x}'\lambda w.\Phi\langle\varphi(\vec{x})(w)\rangle$ and that φ''' reduces to $\lambda\vec{x}'\lambda w.\Phi\langle\varphi'(\vec{x})(w)\rangle$, where w does not occur in Φ. Hence $\varphi''(\vec{\psi}')(\hat{}[\alpha(\vec{\chi})])$ reduces to $[\Phi\langle\varphi(\vec{x})(\hat{}[\alpha(\vec{\chi})])\rangle][\vec{x}' := \vec{\psi}']$, while $\varphi'''(\vec{\psi}')(\hat{}[\alpha'(\vec{\chi})])$ reduces to $[\Phi\langle\varphi'(\vec{x})(\hat{}[\alpha(\vec{\chi})])\rangle][\vec{x}' := \vec{\psi}']$. But since $\varphi(\vec{\psi})(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$ for all $\vec{\psi}_{\vec{a}}$, these are equivalent terms. □

(9) Let $\varphi \Rightarrow_{\mathrm{AR}_{i,f}} \varphi''_{(\vec{a},(((b;f);f);d))}$ and $\varphi' \Rightarrow_{\mathrm{AR}_{i,f}} \varphi'''_{(\vec{a},(((b';f);f);d))}$, and let $\alpha \Rightarrow_{\$} \alpha''_{(\vec{c}',((b;f);f))}$ and $\alpha' \Rightarrow_{\$} \alpha'''_{(\vec{c}',((b';f);f))}$, where \$ consists of $\mathrm{VR}_{j,f}$ possibly followed by AR^{μ}_{k} for $k < j$. Then for all $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}'_{\vec{c}'}$: $\varphi''(\vec{\psi})(\hat{}[\alpha''(\vec{\chi}')]) \Leftrightarrow \varphi'''(\vec{\psi})(\hat{}[\alpha'''(\vec{\chi}')])$.

Proof: observe that φ'' and α'' reduce to $\lambda\vec{x}\lambda w\lambda\vec{x}'.\check{}w(\hat{}\lambda z.\varphi(\vec{x})(z)(\vec{x}'))$ and $\lambda\vec{y}'\lambda v.\mathrm{A}\langle\check{}v(\hat{}[\alpha(\vec{y})])\rangle$, respectively, and that φ''' and α''' reduce to the respective terms $\lambda\vec{x}\lambda w\lambda\vec{x}'.\check{}w(\hat{}\lambda z.\varphi'(\vec{x})(z)(\vec{x}'))$ and $\lambda\vec{y}'\lambda v.\mathrm{A}\langle\check{}v(\hat{}[\alpha'(\vec{y})])\rangle$, where v does not occur in A. Consequently, $\varphi''(\vec{\psi})(\hat{}[\alpha''(\vec{\chi}')])$ reduces to $\lambda\vec{x}'.[\mathrm{A}\langle\varphi(\vec{\psi})(\hat{}[\alpha(\vec{y})])(\vec{x}')\rangle[\vec{y}' := \vec{\chi}']]$, while the term $\varphi'''(\vec{\psi})(\hat{}[\alpha'''(\vec{\chi}')])$ reduces to $\lambda\vec{x}'.[\mathrm{A}\langle\varphi'(\vec{\psi})(\hat{}[\alpha'(\vec{y})])(\vec{x}')\rangle[\vec{y}' := \vec{\chi}']]$. But since $\varphi(\vec{\psi})(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$ for all $\vec{\chi}_{\vec{c}}$, these are equivalent terms. □

Next, let $\varphi_{1,(\vec{a},(b;d))}$, $\varphi_{2,(\vec{a},(b;d))}$, $\varphi'_{1,(\vec{a},(b';d))}$, $\varphi'_{2,(\vec{a},(b';d))}$, $\alpha_{(\vec{c},b)}$, $\alpha'_{(\vec{c},b')}$ be such that for all $\vec{\psi}_{\vec{a}}$, $\vec{\chi}_{\vec{c}}$: $\varphi_1(\vec{\psi})(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \varphi'_1(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$ and $\varphi_2(\vec{\psi})(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \varphi'_2(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$.

(10) If d is a conjoinable type $(\vec{e}, t)$, then for all $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}_{\vec{c}}$: $\mathrm{GC}(\varphi_1, \varphi_2)(\vec{\psi})(\hat{}[\alpha(\vec{\chi})]) \Leftrightarrow \mathrm{GC}(\varphi'_1, \varphi'_2)(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$.

Proof: it suffices to note that $\mathrm{GC}(\varphi_1, \varphi_2)(\vec{\psi})(\hat{}[\alpha(\vec{\chi})])$ reduces to the term $\lambda\vec{z}_{\vec{e}}.[\varphi_1(\vec{\psi})(\hat{}[\alpha(\vec{\chi})])(\vec{z}) \wedge \varphi_2(\vec{\psi})(\hat{}[\alpha_1(\vec{\chi})])(\vec{z})]$, while $\mathrm{GC}(\varphi'_1, \varphi'_2)(\vec{\psi})(\hat{}[\alpha'(\vec{\chi})])$ reduces to the term $\lambda\vec{z}_{\vec{e}}.[\varphi'_1(\vec{\psi})(\hat{}[\alpha(\vec{\chi})])(\vec{z}) \wedge \varphi'_2(\vec{\psi})(\hat{}[\alpha(\vec{\chi})])(\vec{z})]$. □

Finally, let $\varphi_{(\vec{a},(b;d))}$, $\varphi'_{(\vec{a},(b';d))}$, $\alpha_{1,(\vec{c},b)}$, $\alpha_{2,(\vec{c},b)}$, $\alpha'_{1,(\vec{c},b')}$, $\alpha'_{2,(\vec{c},b')}$ be such that for all $\vec{\psi}_{\vec{a}}$, $\vec{\chi}_{\vec{c}}$: $\varphi(\vec{\psi})(\hat{}[\alpha_1(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})(\hat{}[\alpha'_1(\vec{\chi})])$ and $\varphi(\vec{\psi})(\hat{}[\alpha_2(\vec{\chi})]) \Leftrightarrow \varphi'(\vec{\psi})(\hat{}[\alpha'_2(\vec{\chi})])$.

(11) Let $\varphi \Rightarrow_{\mathrm{AR}_{i,t}} \varphi''_{(\vec{a},(((b;t);t);d))}$, $\varphi' \Rightarrow_{\mathrm{AR}_{i,t}} \varphi'''_{(\vec{a},(((b';t);t);d))}$ and for $\gamma \in \{\alpha, \alpha'\}$ and $k \in \{1,2\}$: $\gamma_k \Rightarrow_{\mathrm{VR}_{j,t}} \gamma''_k$. Then for all $\vec{\psi}_{\vec{a}}$ and $\vec{\chi}_{\vec{c}}$: $\varphi''(\vec{\psi})(\hat{}[\mathrm{GC}(\alpha''_1, \alpha''_2)(\vec{\chi})]) \Leftrightarrow \varphi'''(\vec{\psi})(\hat{}[\mathrm{GC}(\alpha'''_1, \alpha'''_2)(\vec{\chi})])$.

Proof: observe that φ'' and φ''' reduce to $\lambda\vec{x}\lambda w\lambda\vec{x}'.{}^{\vee}w(\hat{}\lambda z.\varphi(\vec{x})(z)(\vec{x}'))$ and $\lambda\vec{x}\lambda w\lambda\vec{x}'.{}^{\vee}w(\hat{}\lambda z.\varphi'(\vec{x})(z)(\vec{x}'))$, respectively, and that for $k \in \{1,2\}$ and $\gamma \in \{\alpha, \alpha'\}$: $\gamma''_k = \lambda\vec{y}\lambda v.{}^{\vee}v(\hat{}[\gamma_k(\vec{y})])$. So $\varphi''(\vec{\psi})(\hat{}[\mathrm{GC}(\alpha''_1, \alpha''_2)(\vec{\chi})]) \Leftrightarrow \lambda\vec{x}'.[\varphi(\vec{\psi})(\hat{}[\alpha_1(\vec{\chi})])(\vec{x}') \wedge \varphi(\vec{\psi})(\hat{}[\alpha_2(\vec{\chi})])(\vec{x}')]$ and $\varphi'''(\vec{\psi})(\hat{}[\mathrm{GC}(\alpha'''_1, \alpha'''_2)(\vec{\chi})]) \Leftrightarrow \lambda\vec{x}'.[\varphi'(\vec{\psi})(\hat{}[\alpha'_1(\vec{\chi})])(\vec{x}') \wedge \varphi'(\vec{\psi})(\hat{}[\alpha'_2(\vec{\chi})])(\vec{x}')]$. □

Using (8) through (11), we can extend the scope of the basic equivalences in (5) through (7). Taking for instance (5) as our starting point, we can replace a final application of NR_i to a functor φ which is 'answered' by an application of $\mathrm{V(A)R}_j$ to the $(j-1)$-ary functor α of the i-th argument of φ by one or more applications of AR_i. But with the help of (8) and (9), this can also be done to non-final applications of NR_i to φ. On top of this, (8) and (9) can also be combined with (6) to minimalize the exponents of the resulting applications of AR_i. Going from (functors of) arguments to functors and (using (10) and (11)) from coordinates to coordinations, it is thus possible to transform the semantic structure of every expression into an equivalent one in which every functor φ that combines with an i-th argument with a $(j-1)$-ary functor α only undergoes applications of AR^{μ}_j which correspond to applications of VR^{μ}_j to α.

Moreover, we can in addition exploit specific properties of the fragment. For instance, the fact that all lexical expressions of a non-atomic type are of a type $(\vec{a}, t)$. This means that all applications of AR to lexical translations will have exponent t. Hence applications of VR to functors of their arguments will also have this exponent, so that applications of AR to value-raised translations again have exponent t. Besides, the only expressions which may necessitate an application of AR to their functor for lexical reasons (quantified noun phrases) require exponent t. Summing up, this results in the following candidate type shifts for non-main functor and argument expressions in the fragment (in addition to AL, which is lexically applicable to only two expressions): $\mathrm{VR}_{1,t}$ for expressions which belong to the argument

categories of the fragment (S, T and CN); $AR_{1,t}$ and $VR_{2,t}$ for expressions which belong to unary functor categories (IV and Det); and $AR_{1,t}$, $AR_{2,t}$ and $VR_{3,t}$ for expressions which belong to binary functor categories (TV and PV). Finally, observe that $c \Rightarrow t$ iff $c = t$ (since all type change leads to compound types), so if we are interested in readings (translations of type t) of sentences, we know that the main n-ary functor of the expression cannot have undergone VR_{n+1}. Hence in sentence readings unary main functor expressions (IV) only undergo $AR_{1,t}$, and binary main functor expressions (TV, PV) only undergo $AR_{1,t}$ and $AR_{2,t}$.

Likewise, starting with (7) we know that if a functor φ combines with an i-th argument with a $(j-1)$-ary functor α such that a *final* application of VR_j to α 'answers' a final application of AR_i to φ, both applications can be omitted. However, using (8) we know that the application of AR_i does not necessarily have to be final. And combining (8) and (9) with (7) we can show that if the derivation of α contains two *consecutive* applications of VR_j which correspond to two – possibly non-consecutive – applications of AR_i to φ, we can omit one of the VR_j and the 'innermost' application of AR_i. Similarly, it can be shown that if the derivation of φ contains two *consecutive* applications of AR_i which correspond to two – possibly non-consecutive – applications of VR_i to α, we can omit one of the AR_i and the 'outermost' application of VR_i. Due to (10) and (11), these claims generalize to cases where the translations φ and α consists of (possibly iterated) *coordinations* of translations $\varphi_1, \ldots \varphi_n$ and $\alpha_1, \ldots \alpha_m$ (see (*v*) through (*vii*) below).

Given this much, we can check the claims about numbers of readings for the sentences treated above. We have the following 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 $VR_{1,t}$ to translations of expressions of category S, T, and CN; $AR_{1,t}$, $AR_{2,t}$ and $VR_{3,t}$ to translations of TVs and PVs; and $AR_{1,t}$ and $VR_{1,t}$ to IVs and Dets; (*iv*) apply no VR to (coordinates of) the main functor of the sentence; (*v*) do no final applications of VR_j to the $j-1$-ary main functor of an i-th argument which are 'answered' by an application of AR_i to (the coordinates of) its functor; (*vi*) do no consecutive applications of VR_i to (the coordinates of) the $j-1$-ary main functor of an i-th argument which are 'answered' by two applications of AR_i to (the coordinates of) its functor; (*vii*) do no consecutive applications of AR_i to (the coordinates of) a functor which are 'answered' by two applications of VR_j to (the coordinates of) the

$j-1$-ary main functor of its i-th argument. Consider the following examples:

(12) Mary walks and seeks every unicorn

(13) 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 coordation *walks and seeks every unicorn*. The verb *seeks* can undergo AL_1. The proper name only gives rise to final VR_1 applications which have to be answered by AR_2 (applied to both conjoined verb phrases or to their conjunction), so no type change is needed here (cf. (v)). The quantifier *every unicorn* necessitates AR_1 for lowered *seeks*. VR_1 for this noun phrase would be 'final' and 'answered' – by AR_1 on *seeks* –, hence the verb gets at most one AR_1. For the conjoined verb phrases and their conjunction, VR is ruled out (due to (iv)). We saw that AR_2 on the conjoined verb phrases or their conjunction would have to be answered by a final application of VR_1 to the proper name. So the choice is between $\langle\rangle$ (doing nothing) and $\langle AL_1, AR_1\rangle$ for *seeks*: there are two readings.

Sentence (13) 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 VR_1 applications which have to be answered by their functors *claims* and *wrote* – hence no type change is needed here (see (v)). However, the quantified noun phrases *every schoolboy* and *a mathematician* necessitate ('unanswered') applications of AR_2 in their functors *believes* and *wrote*, respectively. Iteration of AR_2 on these functors would involve applications of final VR_1 to the quantifiers (v). For *wrote*, then, we have exactly one AR_2 and zero AR_1. The remaining option, VR_3, is necessarily answered; hence it must be non-final (on account of (v)) and non-consecutive (due to (vi)). So, we are left with (α) $\langle AR_2\rangle$ and (β) $\langle VR_3, AR_2\rangle$. For *claims*, we have zero AR_2. Besides, VR_3 is impossible for this main functor of the sentence: value raising would not lead to a translation of type t (iv). AR_1 will be answered; hence it must be non-consecutive (vii). I.e., (a) $\langle\rangle$ and (b) $\langle AR_1\rangle$ remain. For *believes*, we have exactly one AR_2 and two possibilities, AR_1 and VR_3. The facts about *wrote* entail that AR_1 will be applied at most once. VR_3 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) $\langle \mathrm{AR}_2 \rangle$; (B) $\langle \mathrm{AR}_1, \mathrm{AR}_2 \rangle$; (C) $\langle \mathrm{AR}_2, \mathrm{AR}_1 \rangle$; (D) $\langle \mathrm{VR}_3, \mathrm{AR}_2 \rangle$; (E) $\langle \mathrm{AR}_2, \mathrm{VR}_3, \mathrm{AR}_1 \rangle$; (F) $\langle \mathrm{AR}_1, \mathrm{VR}_3, \mathrm{AR}_2 \rangle$; (G) $\langle \mathrm{VR}_3, \mathrm{AR}_2, \mathrm{AR}_1 \rangle$; and (H) $\langle \mathrm{VR}_3, \mathrm{AR}_1, \mathrm{AR}_2 \rangle$. 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: there are eight readings.

Chapter 2

Compositionality and Flexibility

> It is to be expected, then, that the aim of syntax could be realized in many different ways, only some of which would provide a suitable basis for semantics.
>
> Richard Montague, *Universal Grammar*

The present chapter is concerned with determining whether the flexible Montague grammar defined in Chapter 1 observes the principle of compositionality. The chapter is structured as follows. In the first section, 'Interpretation and Translation', a small fragment of non-flexible Montague grammar will be presented. Properties of that fragment will play an illustrative role in the second section, 'Compositionality', where we will consider the implications of the principle of compositionality for the organization of grammar fragments in general. In the third section, 'Similarity', these implications are compared to the formalization of the principle presented in Janssen (1986). The conclusion of this comparison is drawn in the fourth section, 'Interpretability': the alternative, somewhat different formalization given in Section 2 can be motivated and applied more easily, but it also avoids the technical complications that encumber Janssen's formalization. In the fifth and final section, 'Flexible Interpretation', we will show that the grammar of Chapter 1 can be readily reformulated as an orthodox compositional grammar in Janssen's 'most intuitive' interpretation of the principle. The necessary and sufficient steps involve a more fine-grained definition of syntactic categories plus a simple reformulation of the semantic 'type-changing' translation derivations in terms of unary syntactic/semantic rules.

1 Interpretation and Translation

In order to provide ourselves with some illustration material for the issues to be discussed in the next sections, we start with a very small Montague grammar fragment of English. For expository purposes, the semantics has been kept extensional. The details of our exposition may sometimes deviate

from what the reader is accustomed to, but we have chosen our set-up so as to facilitate our discussion in the next sections.

First, we define the syntax and semantics of L, a language of extensional typed logic:

(1) T, the set of *types*, is the smallest set such that
(i) $e \in T$ and $t \in T$; and
(ii) if $\sigma \in T$ and $\tau \in T$, then $(\sigma, \tau) \in T$.

For every type $\tau \in T$, there are two sets of basic expressions: (i) a set V_τ consisting of the *variables of type* τ, and (ii) a set C_τ consisting of the *constants of type* τ:

(2) (i) $V_\tau = \{v_{i,\tau} \mid i \in \mathbb{N}\}$;
(ii) $C_e = \{m, j\}$,
$C_{(e,t)} = \{\text{MAN,CONVERSATION,WALK,TALK}\}$,
$C_{((e,t),(e,t))} = \{\text{FAST}\}$, and
$C_\tau = \emptyset$ for $\tau \notin \{e,\ (e,t),\ ((e,t),(e,t))\}$.

The variables $v_{0,e}$, $v_{0,(e,t)}$ and $v_{1,(e,t)}$ will be written as x, P and Q, respectively. For every type $\tau \in T$, the set L_τ of *well-formed expressions of type* τ is the smallest set such that:

(3) (i) $V_\tau \subseteq L_\tau$ and $C_\tau \subseteq L_\tau$;
(ii) if $\varphi \in L_t$, then $F_1(\varphi) \in L_t$;
(iii) if $\varphi \in L_t$ and $\psi \in L_t$, then $F_2(\varphi, \psi) \in L_t$;
(iv) if $\varphi \in L_t$, then $F_{3:\tau:i}(\varphi) \in L_t$;
(v) if $\alpha \in L_{(\sigma,\tau)}$ and $\beta \in L_\sigma$, then $F_{4:\sigma:\tau}(\alpha, \beta) \in L_\tau$;
(vi) if $\alpha \in L_\tau$, then $F_{5:\sigma:\tau:i}(\alpha) \in L_{(\sigma,\tau)}$;
(vii) if $\alpha \in L_\tau$ and $\beta \in L_\tau$, then $F_{6:\tau}(\alpha, \beta) \in L_t$.

The functions involved are defined in (4), where $\sigma \in T$, $\tau \in T$ and $i \in \mathbb{N}$:

(4) (i) $F_1 : L_t \to L_t$, where $F_1(\varphi) = \neg\, \varphi$;
(ii) $F_2 : L_t \times L_t \to L_t$, where $F_2(\varphi, \psi) = [\varphi \wedge \psi]$;
(iii) $F_{3:\tau:i} : L_t \to L_t$, where $F_{3:\tau:i}(\varphi) = \forall v_{i,\tau}\ \varphi$;
(iv) $F_{4:\sigma:\tau} : L_{(\sigma,\tau)} \times L_\sigma \to L_\tau$, where $F_{4:\sigma:\tau}(\alpha, \beta) = [[\alpha](\beta)]$;
(v) $F_{5:\sigma:\tau:i} : L_\tau \to L_{(\sigma,\tau)}$, where $F_{5:\sigma:\tau:i}(\alpha) = \lambda v_{i,\sigma}\ \alpha$;
(vi) $F_{6:\tau} : L_\tau \times L_\tau \to L_t$, where $F_{6:\tau}(\alpha, \beta) = [\alpha = \beta]$.

Other connectives and operators can be considered abbreviations in the usual way. Thus $[\varphi \rightarrow \psi]$ abbreviates $\neg\,[\varphi \wedge \neg\psi] = F_1(F_2(\varphi, F_1(\psi)))$, $\exists v_{i,\tau}\,\varphi$ abbreviates $\neg\,\forall v_{i,\tau}\,\neg\,\varphi = F_1(F_{3:\tau:i}(F_1(\varphi)))$, etc.[1]

The language L itself is the indexed family consisting of the above sets of well-formed expressions, i.e.:

(5) $\quad L = (L_\tau)_{\tau \in T}$

An *interpretation* of L is based on some non-empty set E, the *domain of individuals*. On the basis of E, we define for every $\tau \in T$ a *domain* $D_{E,\tau}$:

(6) $\quad (i) \quad D_{E,e} = E;$
$\quad (ii) \quad D_{E,t} = \{0,1\}$; and
$\quad (iii) \quad D_{E,(\sigma,\tau)} = D_{E,\tau}^{D_{E,\sigma}}$.

The expression $D_{E,\tau}^{D_{E,\sigma}}$ denotes the set of functions from $D_{E,\sigma}$ to $D_{E,\tau}$. Furthermore, a *frame* F is a family of domains indexed by the types, i.e.:

(7) $\quad F = (D_{E,\tau})_{\tau \in T}$

A *constant interpretation* is a function I having the set of all constants as its domain such that if $c \in C_\tau$, then $I(c) \in D_{E,\tau}$.

A *variable assignment* is a function a having the set of all variables as its domain such that for all $v_{i,\tau}$: $a(v_{i,\tau}) \in D_{E,\tau}$. We write $a[v_{i,\tau}/d]$ for the assignment a' such that (i) $a'(v_{i,\tau}) = d$ and (ii) $a'(v_{i',\tau'}) = a(v_{i',\tau'})$ if $i' \neq i$ or $\tau' \neq \tau$. The set of all variable assignments will be denoted by A.

A *model* m is an ordered pair consisting of a frame and a constant interpretation:

(8) $\quad m = \langle (D_{E,\tau})_{\tau \in T}, I \rangle$

Finally, $in_m(\alpha)$, the *interpretation* in m of a well-formed expression α of type τ, is the following function from variable assignments to members of

[1] The present choice is not minimal, by the way. The semantic effects of F_1,F_2 and $F_{3:\tau:i}$ (cf. (10)) can in their turn be defined in terms of $F_{4:\sigma:\tau}$, $F_{5:\sigma:\tau:i}$ and $F_{6:\tau}$. See the Introduction, Henkin (1963), and Gallin (1975).

the set $D_{E,\tau}$:

(9)
- (*i*) if $\alpha \in V_\tau$, then $in_m(\alpha) = \{\langle a, d\rangle \mid a \in A \text{ and } d = a(\alpha)\}$;
- (*ii*) if $\alpha \in C_\tau$, then $in_m(\alpha) = \{\langle a, d\rangle \mid a \in A \text{ and } d = I(\alpha)\}$;
- (*iii*) $in_m(F_1(\varphi)) = G_{m,1}(in_m(\varphi))$;
- (*iv*) $in_m(F_2(\varphi, \psi)) = G_{m,2}(in_m(\varphi), in_m(\psi))$;
- (*v*) $in_m(F_{3:\tau:i}(\varphi)) = G_{m,3:\tau:i}(in_m(\varphi))$;
- (*vi*) $in_m(F_{4:\sigma:\tau}(\alpha, \beta)) = G_{m,4:\sigma:\tau}(in_m(\alpha), in_m(\beta))$;
- (*vii*) $in_m(F_{5:\sigma:\tau:i}(\alpha)) = G_{m,5:\sigma:\tau:i}(in_m(\alpha))$;
- (*viii*) $in_m(F_{6:\tau}(\alpha, \beta)) = G_{m,6:\tau}(in_m(\alpha), in_m(\beta))$.

The functions introduced in (9) are defined in (10):

(10)
- (*i*) $G_{m,1} : D^A_{E,t} \to D^A_{E,t}$, where $G_{m,1}(\delta) =$
  $\{\langle a, 1\rangle \mid \langle a, 0\rangle \in \delta\} \cup$
  $\{\langle a, 0\rangle \mid \langle a, 1\rangle \in \delta\}$;
- (*ii*) $G_{m,2} : D^A_{E,t} \times D^A_{E,t} \to D^A_{E,t}$, where $G_{m,2}(\delta, \varepsilon) =$
  $\{\langle a, 1\rangle \mid \langle a, 1\rangle \in \delta \text{ and } \langle a, 1\rangle \in \varepsilon\} \cup$
  $\{\langle a, 0\rangle \mid \langle a, 0\rangle \in \delta \text{ or } \langle a, 0\rangle \in \varepsilon\}$;
- (*iii*) $G_{m,3:\tau:i} : D^A_{E,t} \to D^A_{E,t}$, where $G_{m,3:\tau:i}(\delta) =$
  $\{\langle a, 1\rangle \mid \text{for every } d \in D_{E,\tau}\text{: } \langle a[v_{i,\tau}/d], 1\rangle \in \delta\} \cup$
  $\{\langle a, 0\rangle \mid \text{for some } d \in D_{E,\tau}\text{: } \langle a[v_{i,\tau}/d], 0\rangle \in \delta\}$;
- (*iv*) $G_{m,4:\sigma:\tau} : D^A_{E,(\sigma,\tau)} \times D^A_{E,\sigma} \to D^A_{E,\tau}$, where $G_{m,4:\sigma:\tau}(\delta, \varepsilon) =$
  $\{\langle a, f(d)\rangle \mid \langle a, f\rangle \in \delta \text{ and } \langle a, d\rangle \in \varepsilon\}$;
- (*v*) $G_{m,5:\sigma:\tau:i} : D^A_{E,\tau} \to D^A_{E,(\sigma,\tau)}$, where $G_{m,5:\sigma:\tau:i}(\delta) =$
  $\{\langle a, f\rangle \mid \text{for every } d \in D_{E,\sigma}\text{: } \langle a[v_{i,\sigma}/d], f(d)\rangle \in \delta\}$;
- (*vi*) $G_{m,6:\tau} : D^A_{E,\tau} \times D^A_{E,\tau} \to D^A_{E,t}$, where $G_{m,6:\tau}(\delta, \varepsilon) =$
  $\{\langle a, 1\rangle \mid \langle a, d\rangle \in \delta \text{ and } \langle a, d'\rangle \in \varepsilon \text{ and } d = d'\} \cup$
  $\{\langle a, 0\rangle \mid \langle a, d\rangle \in \delta \text{ and } \langle a, d'\rangle \in \varepsilon \text{ and } d \neq d'\}$

The above semantics is different from (but equivalent to)[2] the more usual format, in which every well-formed expression α of type τ is assigned an object in $D_{E,\tau}$ *under a variable assignment* a, viz.:

(11) $|\alpha|^{m,a}$

This different set-up is necessitated by the fact that there is not in general a single object in $D_{E,\tau}$ that serves as the denotation of a well-formed

[2]In the sense that $\langle a, d\rangle \in in_m(\alpha)$ if and only if $d = |\alpha|^{m,a}$. See the Introduction for the definition of $|\alpha|^{m,a}$.

expression α of type τ. If α contains free variables, then its denotation is dependent on the interpretation of those free variables and, hence, on the variable assignment. In order to be able to talk about 'the' interpretation of an expression α in a model m, one has to incorporate the variable assignment into the concept of interpretation (cf. Janssen (1986) Part 1, pp. 28–35; Montague (1970b), p. 228).

We will now define a minute fragment of English which will receive its interpretation via expressions of L.

The expressions of the fragment belong to syntactic *categories*, and each category c is associated with a type of L, $\sigma(c)$:

(12)

c	DESCRIPTION	$\sigma(c)$
S	sentence	t
T	term (noun phrase)	$((e,t),t)$
CN	common noun	(e,t)
V	verb	(e,t)
IV	intransitive verb phrase	(e,t)
Adv	adverb	$((e,t),(e,t))$
Det	determiner	$((e,t),((e,t),t))$

Every lexical expression α in category c is assigned one *lexical translation*, $lextr(\alpha)$, a well-formed expression of L of type $\sigma(c)$. Here are the lexical expressions, their categories and their lexical translations:[3]

(13)

α	CATEGORY OF α	$lextr(\alpha)$
man	CN	MAN
talk	CN	CONVERSATION
walk	V	WALK
talk	V	TALK
Mary	T	$\lambda P\ [[P](m)]$
Every man	T	$\lambda P\ [[P](j)]$
Every	Det	$\lambda P\ \lambda Q\ \forall x\ \neg\ [[[P](x)] \wedge \neg[[Q](x)]]$
fast	Adv	FAST

The *syntax* of the fragment is a definition of two families of sets: $(S_c)_{c\in C}$ and $(E_c)_{c\in C}$, where $C = \{\text{S,CN,V,IV,T,Det,Adv}\}$. S_c is the set of *syntactic*

[3]The translation of *Every* can be abbreviated as $\lambda P\ \lambda Q\ \forall x\ [[[P](x)] \rightarrow [[Q](x)]]$ (cf. the comments on (4) above).

terms of category c, and E_c is the set of *English expressions* of category c. For $c \in C$, the sets S_c and E_c are defined as the smallest sets such that:

(14) (*i*) if α of category c occurs in (13), then $\lfloor \alpha_c \rfloor \in S_c$;
(*ii*) if $\alpha \in S_{\mathrm{V}}$, then $\lfloor \underline{H}_1\, \alpha \rfloor \in S_{\mathrm{IV}}$;
(*iii*) if $\alpha \in S_{\mathrm{T}}$ and $\beta \in S_{\mathrm{IV}}$, then $\lfloor \underline{H}_2\, \alpha\beta \rfloor \in S_{\mathrm{S}}$;
(*iv*) if $\alpha \in S_{\mathrm{Det}}$ and $\beta \in S_{\mathrm{CN}}$, then $\lfloor \underline{H}_3\, \alpha\beta \rfloor \in S_{\mathrm{T}}$;
(*v*) if $\alpha \in S_{\mathrm{IV}}$ and $\beta \in S_{\mathrm{Adv}}$, then $\lfloor \underline{H}_4\, \alpha\beta \rfloor \in S_{\mathrm{IV}}$;
(*vi*) if $\alpha \in S_{\mathrm{S}}$ and $\beta \in S_{\mathrm{S}}$, then $\lfloor \underline{H}_5\, \alpha\beta \rfloor \in S_{\mathrm{S}}$;
(*vii*) if $\alpha \in S_{\mathrm{IV}}$ and $\beta \in S_{\mathrm{IV}}$, then $\lfloor \underline{H}_6\, \alpha\beta \rfloor \in S_{\mathrm{IV}}$.

(15) (*i*) if α of category c occurs in (13), then $\alpha \in E_c$;
(*ii*) if $\alpha \in E_{\mathrm{V}}$, then $H_1(\alpha) \in E_{\mathrm{IV}}$;
(*iii*) if $\alpha \in E_{\mathrm{T}}$ and $\beta \in E_{\mathrm{IV}}$, then $H_2(\alpha, \beta) \in E_{\mathrm{S}}$;
(*iv*) if $\alpha \in E_{\mathrm{Det}}$ and $\beta \in E_{\mathrm{CN}}$, then $H_3(\alpha, \beta) \in E_{\mathrm{T}}$;
(*v*) if $\alpha \in E_{\mathrm{IV}}$ and $\beta \in E_{\mathrm{Adv}}$, then $H_4(\alpha, \beta) \in E_{\mathrm{IV}}$;
(*vi*) if $\alpha \in E_{\mathrm{S}}$ and $\beta \in E_{\mathrm{S}}$, then $H_5(\alpha, \beta) \in E_{\mathrm{S}}$;
(*vii*) if $\alpha \in E_{\mathrm{IV}}$ and $\beta \in E_{\mathrm{IV}}$, then $H_6(\alpha, \beta) \in E_{\mathrm{IV}}$.

Whereas the $\underline{H}_i$ merely function as symbols, the functions H_i are defined as the following operations on strings:

(16) (*i*) $H_1 : E_{\mathrm{V}} \rightarrow E_{\mathrm{IV}}$, where $H_1(\alpha) = \alpha s$;
(*ii*) $H_2 : E_{\mathrm{T}} \times E_{\mathrm{IV}} \rightarrow E_{\mathrm{S}}$, where $H_2(\alpha, \beta) = \alpha\ \beta$;
(*iii*) $H_3 : E_{\mathrm{Det}} \times E_{\mathrm{CN}} \rightarrow E_{\mathrm{T}}$, where $H_3(\alpha, \beta) = \alpha\ \beta$;
(*iv*) $H_4 : E_{\mathrm{IV}} \times E_{\mathrm{Adv}} \rightarrow E_{\mathrm{IV}}$, where $H_4(\alpha, \beta) = \alpha\ \beta$;
(*v*) $H_5 : E_{\mathrm{S}} \times E_{\mathrm{S}} \rightarrow E_{\mathrm{S}}$, where $H_5(\alpha, \beta) = \alpha\ \textit{and}\ \beta$;
(*vi*) $H_6 : E_{\mathrm{IV}} \times E_{\mathrm{IV}} \rightarrow E_{\mathrm{IV}}$, where $H_6(\alpha, \beta) = \alpha\ \textit{and}\ \beta$.

The function ev: $\bigcup_{c \in C} S_c \rightarrow \bigcup_{c \in C} E_c$ evaluates each syntactic term in S_c as an English expression in E_c.

(17) (*i*) $ev(\lfloor \alpha_c \rfloor) = \alpha$;
(*ii*) $ev(\lfloor \underline{H}_1\, \alpha \rfloor) = H_1(ev(\alpha))$;
(*iii*) $ev(\lfloor \underline{H}_2\, \alpha\beta \rfloor) = H_2(ev(\alpha), ev(\beta))$;
(*iv*) $ev(\lfloor \underline{H}_3\, \alpha\beta \rfloor) = H_3(ev(\alpha), ev(\beta))$;
(*v*) $ev(\lfloor \underline{H}_4\, \alpha\beta \rfloor) = H_4(ev(\alpha), ev(\beta))$;
(*vi*) $ev(\lfloor \underline{H}_5\, \alpha\beta \rfloor) = H_5(ev(\alpha), ev(\beta))$;
(*vii*) $ev(\lfloor \underline{H}_6\, \alpha\beta \rfloor) = H_6(ev(\alpha), ev(\beta))$.

Every syntactic term $\alpha \in S_c$ is assigned a translation $tr(\alpha) \in L_{(\sigma(c))}$:

(18) (*i*) $tr(\lfloor \alpha_c \rfloor) = lextr(\alpha)$ for α of category c in (13);
(*ii*) $tr(\lfloor \underline{H}_1\, \alpha \rfloor) = F_x(tr(\alpha))$;
(*iii*) $tr(\lfloor \underline{H}_2\, \alpha\beta \rfloor) = F_{4:(e,t):t}(tr(\alpha), tr(\beta))$;
(*iv*) $tr(\lfloor \underline{H}_3\, \alpha\beta \rfloor) = F_{4:(e,t):((e,t),t)}(tr(\alpha), tr(\beta))$;
(*v*) $tr(\lfloor \underline{H}_4\, \alpha\beta \rfloor) = F_y(tr(\alpha), tr(\beta))$;
(*vi*) $tr(\lfloor \underline{H}_5\, \alpha\beta \rfloor) = F_2(tr(\alpha), tr(\beta))$;
(*vii*) $tr(\lfloor \underline{H}_6\, \alpha\beta \rfloor) = F_z(tr(\alpha), tr(\beta))$.

The functions invoked in (18) are given in (19) (note that the functions $F_{4:(e,t):t}$, $F_{4:(e,t):((e,t),t)}$ and F_2 have already been defined in (4) above):

(19) (*i*) $F_x : L_{(e,t)} \rightarrow L_{(e,t)}$,
where $F_x(\alpha) = \alpha$;
(*ii*) $F_{4:(e,t):t} : L_{((e,t),t)} \times L_{(e,t)} \rightarrow L_t$,
where $F_{4:(e,t):t}(\alpha, \beta) = [[\alpha](\beta)]$;
(*iii*) $F_{4:(e,t):((e,t),t)} : L_{((e,t),((e,t),t))} \times L_{(e,t)} \rightarrow L_{((e,t),t)}$,
where $F_{4:(e,t):((e,t),t)}(\alpha, \beta) = [[\alpha](\beta)]$;
(*iv*) $F_y : L_{(e,t)} \times L_{((e,t),(e,t))} \rightarrow L_{(e,t)}$,
where $F_y(\alpha, \beta) = [[\beta](\alpha)]$;
(*v*) $F_2 : L_t \times L_t \rightarrow L_t$,
where $F_2(\alpha, \beta) = [\alpha \wedge \beta]$
(*vi*) $F_z : L_{(e,t)} \times L_{(e,t)} \rightarrow L_{(e,t)}$,
where $F_z(\alpha, \beta) = \lambda x\, [[[\alpha](x)] \wedge [[\beta](x)]]$.

Every English expression α is associated with a set of translations $Tr(\alpha)$. This set consists of the translations of the syntactic terms the evaluation of which coincides with α:

(20) If $\alpha \in E_c$, then $Tr(\alpha) = \{tr(\gamma) \mid \gamma \in S_c \text{ and } ev(\gamma) = \alpha\}$

Finally, interpretation is the composition of the mappings tr and in_m. For a syntactic term $\alpha \in S_c$, the interpretation of α in m is equal to $in_m(tr(\alpha))$. And an English expression $\alpha \in E_c$ can be associated with a set of interpretations in m, viz.: $\{in_m(\beta) \mid \beta \in Tr(\alpha)\}$.

2 Compositionality

In its most general form, the principle of compositionality states that

> the meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined. (Partee (1984), p. 281)

In other words: the meaning of an expression is determined completely by the meanings of its parts plus the information which syntactic rules have been used to build that expression out of those parts. The principle of compositionality is also known as 'Frege's principle', but Janssen (1986) argues that this attribution is at best a tribute. We will give a formalization of the principle along the lines of Janssen (1986), which, in turn, is based on Montague's paper 'Universal Grammar' (1970b).[4]

As for the syntax, the principle presupposes some set A of expressions, and some set F of syntactic rules. In keeping with the customary assumption within theories of formal grammar that linguistic expressions belong to different syntactic *categories*, we will suppose that this set of expressions is an indexed family of sets: $A = (A_s)_{s \in S}$, where S is the set of syntactic categories (or *sorts*), and for each $s \in S$, the set A_s is the set of expressions

[4] Janssen's formalization and the framework defined in Montague's 'Universal Grammar' (UG) are, roughly speaking, 'different views of the same mathematical object' (Janssen (1986), Part 1, p. 91). The main difference is that Janssen employs many-sorted algebras, whereas Montague uses one-sorted algebras (though with much additional structure). As a consequence, Janssen's approach has the following advantages (of which (*a*) through (*c*) are also noted in Janssen (1986), Part 1, pp. 90-92):

(*a*) In UG, the operators in the algebraic sense are untyped, but the syntactic rules are typed. UG requires for each operator a single corresponding semantic operation. However, sometimes one might want to be able to interpret the same operation in different ways, depending on the type of the expressions involved (cf., e.g., the operations H_5 and H_6 of S-conjunction and IV-conjunction which were defined in the previous section).

(*b*) Both frameworks require that the operators be total. In the one-sorted context of UG this means that an algebraic operator has to be defined for all elements of the algebra, also for those elements to which the corresponding syntactic rule will never be applied. And worse, even a semantic interpretation has to be specified for the resulting non-expressions.

(*c*) Janssen's formalization establishes a natural relation between the disambiguated language (the members of the term algebra) and the generated language: one obtains an expression of the generated language from an expression in the term algebra by evaluating the latter. UG only requires some (further unspecified) relation R between the disambiguated language and the generated language.

(*d*) While in Janssen's approach syntactic rules operate on expressions of the generated language, they operate on expressions of the disambiguated language in UG. This restricts the set of possible syntactic operations in an unnatural way. For example, an operation $F : A_s \times A_s \to A_s$ of simple concatenation is not an allowable structural operation in a disambiguated language, since it makes an expression $\alpha\beta\gamma$ ambigous between $F(\alpha, F(\beta, \gamma))$ and $F(F(\alpha, \beta), \gamma)$ (cf. Halvorsen and Ladusaw (1979), footnote 17, p. 221).

of category s, or the *carrier* of sort s. Since we are dealing with expressions, we will assume that the members of the carriers are strings over some alphabet. But there are no further restrictions on the carriers; they may overlap, be empty, include one another, etc.

Syntactic rules (or *operators*) $F_i \in F$ must yield one compound expression e_{n+1} when they are applied to a number of expressions $e_1, \ldots, e_n$, the parts. We will assume that every F_i has a fixed number n of expressions to which it applies (where $n \in \mathbb{N}^+$, i.e., $n \in \mathbb{N}$ and $n > 0$).[5] A syntactic rule does not have to yield a expression for every sequence $\langle e_1, \ldots, e_n \rangle$ of expressions in $\cup A \times \ldots \times \cup A$. It may be that F_i only produces an outcome for sequences $\langle e_1, \ldots, e_n \rangle$ whose components belong to certain categories: $\langle e_1, \ldots, e_n \rangle \in A_{s_1} \times \ldots \times A_{s_n}$. But if this is the case, we assume that F_i does this for *all* $\langle e_1, \ldots, e_n \rangle \in A_{s_1} \times \ldots \times A_{s_n}$, and the outcomes e_{n+1} must without exception belong to the carrier of one sort, $A_{s_{n+1}}$, say.[6] Thus every syntactic operator is a total function $F_i : A_{s_1} \times \ldots \times A_{s_n} \rightarrow A_{s_{n+1}}$, for some $n > 0$, $A_{s_1} \in A$, $\ldots$, $A_{s_n} \in A$, and $A_{s_{n+1}} \in A$. There are no further restrictions on F. The operators may do anything: concatenate, insert, permute, delete, introduce (so-called *syncategorematic*) material that does not occur in the arguments, or what have you.[7]

The picture of syntax which emerges from these considerations is that of a *many-sorted algebra of signature* π:

(21) $\langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ is a *many-sorted algebra* of *signature* π iff
- (*a*) S is a non-empty set (of *sorts*);
- (*b*) $(A_s)_{s \in S}$ is an indexed family of sets (A_s is the *carrier* of s);
- (*c*) Γ is a set (of *operator indices*);
- (*d*) π (the *type-assigning function*) assigns to each $\gamma \in \Gamma$ a pair $\langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$, where $n > 0$, $s_1 \in S$, $\ldots$, $s_{n+1} \in S$; and

[5] Contrary to Montague (1970b), we will follow Janssen (1986) in not allowing infinitary operators. This restriction, which does not seem to reduce the practical applicability of the framework, has the advantage that one will never have to deal with proper classes of operators (cf. Montague (1970b), footnote 4).

[6] This assumption does not involve a real restriction. The effects of operators which do not have a fixed arity, are partial, do not distinguish among argument sorts, or yield values of more than one sort can be mimicked by a suitable adaptation of the structure of F and A: by making it more fine-grained, adding operators, subdividing carriers into different (new) sorts, etc.

[7] Accordingly, Janssen shows that any recursively enumerable language can be generated by an algebraic grammar. Moreover, such a language can be assigned any set of meanings in a compositional way (1986, Part 1, Chapter 2, Section 3 and Section 6). It follows that compositionality is not an empirical principle, but a methodological one.

(e) $(F_\gamma)_{\gamma\in\Gamma}$ is an indexed family (of *operators*) such that if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then $F_\gamma : A_{s_1} \times \ldots \times A_{s_n} \rightarrow A_{s_{n+1}}$.

In the sequel, the phrase 'many-sorted algebra of signature π' will generally be abbreviated as 'π-algebra'.[8]

Consider, by way of example, the language L of extensional typed logic that was defined in the previous section. Together with its syntactic operators, L constitutes a many-sorted algebra $B = \langle(L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ of signature ω:

- T: the set of sorts of B is the set of types defined in (1).
- L_τ: for every $\tau \in T$, the carrier of τ is defined in (3) as the set of well-formed expressions of type τ.
- Γ: the set of operator indices is obtained by collecting the subscripts of the functions listed in (4): $\Gamma = \{1,2\} \cup (3{:}\tau{:}i)_{\tau\in T, i\in\mathbb{N}} \cup (4{:}\sigma{:}\tau)_{\sigma\in T,\tau\in T} \cup (5{:}\sigma{:}\tau{:}i)_{\sigma\in T,\tau\in T,i\in\mathbb{N}} \cup (6{:}\tau)_{\tau\in T}$.
- ω: the type-assigning function is defined as follows:

  $\omega(1) = \langle\langle t\rangle, t\rangle$; $\omega(2) = \langle\langle t,t\rangle, t\rangle$;
  $\omega(3{:}\tau{:}i) = \langle\langle t\rangle, t\rangle$; $\omega(4{:}\sigma{:}\tau) = \langle\langle(\sigma,\tau),\sigma\rangle,\tau\rangle$;
  $\omega(5{:}\sigma{:}\tau{:}i) = \langle\langle\tau\rangle,(\sigma,\tau)\rangle$; $\omega(6{:}\tau) = \langle\langle\tau,\tau\rangle, t\rangle$.

- F_γ: for every $\gamma \in \Gamma$, F_γ is defined in (4).

Another example of a many-sorted algebra is the syntax of a fragment of English in the previous section: $A = \langle(E_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ of signature π, viz.,

- C: the set of sorts of A is given in (12): C = {S,T,V,IV,CN,Det,Adv}.
- E_c: for every $c \in C$, the carrier of c is defined in (15) as E_c, the set of English expressions of category c.
- Δ: the set of operator indices is obtained by collecting the subscripts of the functions defined in (16): $\Delta = \{1,2,3,4,5,6\}$.

[8]The term 'many-sorted algebra' stems from Adj (1977). Our terminology deviates from Janssen (1986), where a pair $\langle(A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ meeting the requirements in (21) is called a 'many-sorted algebra of signature (S,Γ,π)' (1986, Part 1, p. 43).

- π: the type-assigning function is defined as follows:

  $\pi(1) = \langle\langle \mathrm{V}\rangle, \mathrm{IV}\rangle$; $\quad \pi(2) = \langle\langle \mathrm{T}, \mathrm{IV}\rangle, \mathrm{S}\rangle$;
  $\pi(3) = \langle\langle \mathrm{Det}, \mathrm{CN}\rangle, \mathrm{T}\rangle$; $\quad \pi(4) = \langle\langle \mathrm{IV}, \mathrm{Adv}\rangle, \mathrm{IV}\rangle$;
  $\pi(5) = \langle\langle \mathrm{S}, \mathrm{S}\rangle, \mathrm{S}\rangle$; $\quad \pi(6) = \langle\langle \mathrm{IV}, \mathrm{IV}\rangle, \mathrm{IV}\rangle$.

- H_δ: for every $\delta \in \Delta$, H_δ is defined in (16).

A useful notion is that of a *subalgebra*. A subalgebra of a many-sorted algebra A is a collection of subsets of the carriers of A which is closed under the restrictions of the original operations to those subsets:

(22) A *subalgebra* of π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ is a π-algebra $\langle (B_s)_{s\in S}, (F'_\gamma)_{\gamma\in\Gamma}\rangle$, where $B_s \subseteq A_s$ for all $s \in S$, and for $\gamma \in \Gamma$: if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then F'_γ is the *restriction* of F_γ to $(B_s)_{s\in S}$, i.e.: $F'_\gamma = F_\gamma \cap ((B_{s_1} \times \ldots \times B_{s_n}) \times B_{s_{n+1}})$.

Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a π-algebra which *contains* H, that is: $H = (H_s)_{s\in S}$, and $H_s \subseteq A_s$ for every $s \in S$. The smallest subalgebra $\langle (B_s)_{s\in S}, (F'_\gamma)_{\gamma\in\Gamma}\rangle$ of A which contains H is called the *subalgebra generated by* H, which we will write as $\langle [H], (F'_\gamma)_{\gamma\in\Gamma}\rangle$, where $[H]$ indicates the indexed family $(B_s)_{s\in S}$ of carriers of that subalgebra and for $F'_\gamma \in (F'_\gamma)_{\gamma\in\Gamma}$, F'_γ is the restriction of F_γ to $(B_s)_{s\in S}$. The generated subalgebra $\langle [H], (F'_\gamma)_{\gamma\in\Gamma}\rangle$ always exists, since it can be characterized as the intersection of all subalgebras of A which contain H.

Let algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ contain H. Then H is a *generating set* for A (or: A is *generated by* H) iff $\langle [H], (F'_\gamma)_{\gamma\in\Gamma}\rangle = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$.

Observe that the logical algebra B and the English algebra A we just considered can both be characterized as generated algebras:[9]

The logical ω-algebra $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ can be characterized as the generated algebra $\langle [H], (F_\gamma)_{\gamma\in\Gamma}\rangle$, where $H = (V_\tau \cup C_\tau)_{\tau\in T}$, since the family which, for each type τ, contains the union of the sets of variables and constants of type τ is a generating set for B. (The sets V_τ and C_τ are defined in (2) above; that they generate $(L_\tau)_{\tau\in T}$ appears from (3).)

[9]The move from many-sorted algebras to generated many-sorted algebras involves by no means a loss of generality or applicability of the formalization: every many-sorted algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ can be characterized as a generated algebra, since it is trivially true that $A = \langle [(A_s)_{s\in S}], (F_\gamma)_{\gamma\in\Gamma}\rangle$.

The English π-algebra $A = \langle (E_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ can also be characterized as a generated algebra: viz., $A = \langle [K], (H_\delta)_{\delta\in\Delta}\rangle$, where

(23) $K = \{K_{\mathrm{T}}, K_{\mathrm{V}}, K_{\mathrm{S}}, K_{\mathrm{IV}}, K_{\mathrm{CN}}, K_{\mathrm{Det}}, K_{\mathrm{Adv}}\}$,
$K_{\mathrm{T}} = \{\textit{Mary,Every man}\}$, $K_{\mathrm{V}} = \{\textit{walk,talk}\}$,
$K_{\mathrm{S}} = K_{\mathrm{IV}} = \emptyset$, $K_{\mathrm{CN}} = \{\textit{man,talk}\}$,
$K_{\mathrm{Det}} = \{\textit{Every}\}$, $K_{\mathrm{Adv}} = \{\textit{fast}\}$.

There is an important difference between $\langle [H], (F_\gamma)_{\gamma\in\Gamma}\rangle$, the generated algebra for the logical language, and $\langle [K], (H_\delta)_{\delta\in\Delta}\rangle$, the generated algebra for the fragment of English. Whereas the members of the carriers $(L_\tau)_{\tau\in T}$ of $\langle [H], (F_\gamma)_{\gamma\in\Gamma}\rangle$ can be analyzed in a unique way, this does not hold for the members of the carriers $(E_c)_{c\in C}$ of $\langle [K], (H_\delta)_{\delta\in\Delta}\rangle$.

For instance, the expression *Every man* is a member of E_{T} on two accounts: it can be analyzed as a lexical expression (i.e., as a member of K_{T}), but also as the compound expression $H_3(\textit{Every}, \textit{man})$.

Moreover, the lexical expression *talk* is a member of two carriers, E_{V} and E_{CN}: it is both a verb and a common noun.

And finally, the compound expression *Mary walks and talks fast*, a member of E_S, can be analyzed as both (24) and (25):

(24) $H_2(\textit{Mary}, H_6(H_1(\textit{walk}), H_4(H_1(\textit{talk}), \textit{fast})))$

(25) $H_2(\textit{Mary}, H_4(H_6(H_1(\textit{walk}), H_1(\textit{talk})), \textit{fast}))$

In other words, the English algebra $A = \langle [K], (H_\delta)_{\delta\in\Delta}\rangle$ is *syntactically ambiguous*. The logical algebra $B = \langle [H], (F_\gamma)_{\gamma\in\Gamma}\rangle$, on the other hand, is syntactically *un*ambiguous, because:

(*a*) the members of the generating set $(H_\tau)_{\tau\in T}$ are not in the range of some operator F_γ in $(F_\gamma)_{\gamma\in\Gamma}$: if $e \in H_\tau$, then for all F_γ and $e_1, \ldots, e_n$: $e \neq F_\gamma(e_1, \ldots, e_n)$;

(*b*) every expression in $[H] = [(H_\tau)_{\tau\in T}] = (L_\tau)_{\tau\in T}$ is a member of exactly one carrier: if $e \in L_\tau$ and $e \in L_{\tau'}$, then $\tau = \tau'$; and

(*c*) the operators in $(F_\gamma)_{\gamma\in\Gamma}$ are injections which have disjoint ranges: if $F_\gamma(e_1, \ldots, e_n) = F_{\gamma'}(e'_1, \ldots, e'_m)$, then $\langle e_1, \ldots, e_n\rangle = \langle e'_1, \ldots, e'_m\rangle$ and $F_\gamma = F_{\gamma'}$.

If a generated algebra $\langle[(H_\tau)_{\tau\in T}],(F_\gamma)_{\gamma\in\Gamma}\rangle$ has the properties (a) through (c), it is called a *free algebra* generated by $(H_\tau)_{\tau\in T}$.[10]

As a consequence of the phenomenon of syntactic ambiguity, one cannot in general speak of the meaning of an expression,[11] but only of the meaning of an expression with respect to a certain sort and a so-called derivational history.

In order to be able to refer to sorts and derivational histories of expressions, we introduce the notion of a *term algebra*:

(26) If π-algebra $A = \langle(A_s)_{s\in S},(F_\gamma)_{\gamma\in\Gamma}\rangle$ is generated by $H = (H_s)_{s\in S}$, then $T_{A,H}$, the *term algebra of A with respect to H*, is the algebra $\langle(T_{A,H,s})_{s\in S},(F^T_\gamma)_{\gamma\in\Gamma}\rangle$, where for all $s \in S$ and for all $\gamma \in \Gamma$:
(a) $T_{A,H,s}$ is the smallest set such that $\{\lfloor h_s\rfloor \mid h \in H_s\} \subseteq T_{A,H,s}$, and if $t_1 \in T_{A,H,s_1},\ldots,t_n \in T_{A,H,s_n}$ and $\pi(\gamma) = \langle\langle s_1,\ldots,s_n\rangle,s\rangle$,

[10]Clause (a) and (c) constitute the definition of the notion of a 'free algebra' for traditional 'one-sorted' algebras (cf. Montague (1970b), p. 225). Clause (b) seems a natural addition in the context of many-sorted algebras.

After a comparison of his own formalization with the many-sorted approach outlined in Adj (1977), Janssen concludes that he 'did not succeed in obtaining a handsome definition of the notion of a "free algebra" ' (Janssen (1986), Part 1, p. 94), and blames this on the fact that he does not treat members of carriers as nullary operators. We agree with Janssen that such a treatment is 'intuitively difficult, and practically inconvenient' (p. 93), but we do not consider it necessary for a definition of the notion of a 'free algebra': intuitively, the expressions in the carriers of a free algebra should be uniquely analyzable, and this simply disallows expressions which occur in more than one carrier. A related issue is the definition of the notion of a 'homomorphism'. In Janssen's (and our) approach, a homomorphism is a function which has the union of the collection of carriers as its domain (cf. definition (32) below). In Adj (1977), it is a sorted collection of functions: for each sort there is a separate function. Thus if an object occurs in two carriers, these occurrences are treated as two different objects and can be assigned different values by the homomorphism, something which is not acceptable if a homomorphism is a single function (Janssen, (1986), p. 93).

However, this problem is readily solved if one takes into account that homomorphisms (the cornerstones of semantic interpretation) are defined on term algebras. If term algebras are designed so as to represent occurrences of an object in different carriers as different objects (e.g., by subscripting them with the sort of the carrier to which they belong, cf. $\lfloor h_s\rfloor$ in (26)), it is possible to assign different values to these occurrences. Besides, such term algebras will be free in the sense of (a) through (c), as will become clear below.

[11]Correspondingly (see (13) and (18) in the previous section), the lexical expression *Every man* of category T is treated like a proper name (as in the medieval morality play) witness its translation $\lambda P\,[P(j)]$, while its analysis as $H_3(\mathit{Every}, \mathit{man})$ is responsible for a quantifier translation which reduces to $\lambda Q\,\forall x\,[[\textsc{man}(x)] \rightarrow [Q(x)]]$. Similarly, the expression *talk* is assigned two translations: CONVERSATION (as a CN) and TALK (as a V). The analyses (24) and (25) of *Mary walks and talks fast* yield the reduced translations $[[\textsc{walk}(m)] \wedge [[\textsc{fast}(\textsc{talk})](m)]]$ and $[[\textsc{fast}(\lambda x\,[[\textsc{walk}(x)] \wedge [\textsc{talk}(x)]])](m)]$, respectively.

then $F_\gamma^T(t_1,\ldots,t_n) \in T_{A,H,s}$; and
(*b*) $F_\gamma^T(t_1,\ldots,t_n) = \lfloor \underline{F}_\gamma\, t_1 \ldots t_n \rfloor$.

The carriers of a term algebra consist of symbols which can be seen as representations of the derivational histories of a generated algebra.

Note that the term algebra $T_{A,H}$ of a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating set $H = (H_s)_{s\in S}$ can be characterized as the generated algebra $\langle [(\{\lfloor h_s \rfloor \mid h \in H_s\}_s)_{s\in S}], (F_\gamma^T)_{\gamma\in\Gamma}\rangle$, and that the latter is always a free algebra since it meets the requirements (*a*) through (*c*):

(*a*) members $\lfloor h_s \rfloor$ of the generating set are always different from terms $F_\gamma^T(t_1,\ldots,t_n) = \lfloor \underline{F}_\gamma\, t_1 \ldots t_n \rfloor = \lfloor \underline{F}_\gamma \lfloor x_1 \rfloor \ldots \lfloor x_n \rfloor \rfloor$;

(*b*) generators $\lfloor h_s \rfloor$ carry their sort as a subscript; moreover, if $\pi(\gamma) = \langle \langle s_1,\ldots,s_n\rangle, s_{n+1}\rangle$, then $F_\gamma^T(t_1,\ldots,t_n)$ belongs to sort s_{n+1};

(*c*) $F_\gamma^T(t_1,\ldots,t_n) = \lfloor \underline{F}_\gamma\, t_1 \ldots t_n \rfloor$ for all F_γ^T and $t_1,\ldots,t_n$. Therefore, if $F_\gamma^T(t_1,\ldots,t_n) = F_{\gamma'}^T(t'_1,\ldots,t'_m)$, then $\lfloor \underline{F}_\gamma\, t_1 \ldots t_n \rfloor = \lfloor \underline{F}_{\gamma'}\, t'_1 \ldots t'_m \rfloor$, i.e., $\langle t_1,\ldots,t_n\rangle = \langle t'_1,\ldots,t'_m\rangle$ and $\underline{F}_\gamma = \underline{F}_{\gamma'}$, so that $F_\gamma^T = F_{\gamma'}^T$.

In (14) above, we defined so-called 'syntactic terms'. Those terms are the members of the carriers of the term algebra $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H_\delta^T)_{\delta\in\Delta}\rangle$ of the English algebra $A = \langle (E_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ with respect to the generating set K that was given in (23), where $S_c = T_{A,K,c}$ for all $c \in C$.

Definition (17) relates the members of the carriers of this term algebra $T_{A,K}$ to the members of the carriers of the original algebra A by means of the function *ev*. For example:

(27) $ev(\lfloor \textit{Every man}_{\mathrm{T}} \rfloor) = ev(\lfloor \underline{H}_3 \lfloor \textit{Every}_{\mathrm{Det}} \rfloor \lfloor \textit{man}_{\mathrm{CN}} \rfloor \rfloor) = \textit{Every man}$;

$ev(\lfloor \textit{talk}_{\mathrm{V}} \rfloor) = ev(\lfloor \textit{talk}_{\mathrm{CN}} \rfloor) = \textit{talk}$; and

$ev(\lfloor \underline{H}_2 \lfloor \textit{Mary}_{\mathrm{T}} \rfloor \lfloor \underline{H}_6 \lfloor \underline{H}_1 \lfloor \textit{walk}_{\mathrm{V}} \rfloor \rfloor \lfloor \underline{H}_4 \lfloor \underline{H}_1 \lfloor \textit{talk}_{\mathrm{V}} \rfloor \rfloor \lfloor \textit{fast}_{\mathrm{Adv}} \rfloor \rfloor \rfloor) =$
$ev(\lfloor \underline{H}_2 \lfloor \textit{Mary}_{\mathrm{T}} \rfloor \lfloor \underline{H}_4 \lfloor \underline{H}_6 \lfloor \underline{H}_1 \lfloor \textit{walk}_{\mathrm{V}} \rfloor \rfloor \lfloor \underline{H}_1 \lfloor \textit{talk}_{\mathrm{V}} \rfloor \rfloor \rfloor \lfloor \textit{fast}_{\mathrm{Adv}} \rfloor \rfloor \rfloor) =$
Mary walks and talks fast.

This evaluation function can be defined for arbitrary term algebras:

(28) $ev(\lfloor h_s \rfloor) = h$, and
$ev(\lfloor \underline{F}_\gamma\, t_1 \ldots t_n \rfloor) = F_\gamma(ev(t_1),\ldots,ev(t_n))$.

Note that if $T_{A,H}$ is the term algebra of an algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with respect to some generating set H, then $a \in A_s$ if and only if there is a term $t \in T_{A,H,s}$ such that $ev(t) = a$.

The operators of the English term algebra $T_{A,K}$ form the set $(H_\delta^T)_{\delta\in\Delta}$, where $\Delta = \{1,2,3,4,5,6\}$. The members of this set are defined as follows:

(29)
(i) $H_1^T\colon T_{A,K,\mathrm{V}} \to T_{A,K,\mathrm{IV}}$; $H_1^T(\alpha) = \lfloor \underline{H}_1\, \alpha \rfloor$
(ii) $H_2^T\colon T_{A,K,\mathrm{T}} \times T_{A,K,\mathrm{IV}} \to T_{A,K,\mathrm{S}}$; $H_2^T(\alpha,\beta) = \lfloor \underline{H}_2\, \alpha\beta \rfloor$
(iii) $H_3^T\colon T_{A,K,\mathrm{Det}} \times T_{A,K,\mathrm{CN}} \to T_{A,K,\mathrm{T}}$; $H_3^T(\alpha,\beta) = \lfloor \underline{H}_3\, \alpha\beta \rfloor$
(iv) $H_4^T\colon T_{A,K,\mathrm{IV}} \times T_{A,K,\mathrm{Adv}} \to T_{A,K,\mathrm{IV}}$; $H_4^T(\alpha,\beta) = \lfloor \underline{H}_4\, \alpha\beta \rfloor$
(v) $H_5^T\colon T_{A,K,\mathrm{S}} \times T_{A,K,\mathrm{S}} \to T_{A,K,\mathrm{S}}$; $H_5^T(\alpha,\beta) = \lfloor \underline{H}_5\, \alpha\beta \rfloor$
(vi) $H_6^T\colon T_{A,K,\mathrm{IV}} \times T_{A,K,\mathrm{IV}} \to T_{A,K,\mathrm{IV}}$; $H_6^T(\alpha,\beta) = \lfloor \underline{H}_6\, \alpha\beta \rfloor$

Term algebras play an important role in the formalization of the compositionality principle. Thus, since the meaning of an expression depends on its sort and derivational history (which are represented in the term algebra), the meanings of the members of the carriers of the generated algebra of English expressions $A = \langle [(K_c)_{c\in C}], (H_\delta)_{\delta\in\Delta}\rangle$ are defined on the members of the carriers of the corresponding term algebra $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H_\delta^T)_{\delta\in\Delta}\rangle$.

What about these meanings? By analogy with the idea that linguistic expressions belong to different categories, it has become customary to assume that their interpretations, or meanings, inhabit various semantic *types*, and that, moreover, the interpretations of different expressions of the same category c belong to the same semantic type.

First of all, we will therefore assume that also the members of the semantic domain constitute a family of sets indexed by sorts (the types), the carriers of which contain the semantic objects that can serve as interpretations of syntactic expressions, and that, furthermore, there is a function σ which associates every syntactic category c with its semantic type $\sigma(c)$.

Secondly, given this much, the principle of compositionality – according to which the meaning of a compound expression is a *function* of the meanings of its constituent parts and the way they are syntactically combined[12] – literally requires that for every way of syntactically combining expressions (i.e., for every syntactic operator F_i) there is a semantic *function* G_j such that for every sequence $e_1, \ldots, e_n$ of expressions, the meaning of the compound

[12] I.e., the meaning of an expression relativized to a particular sort and derivational history of that expression.

expression built up from these expressions (i.e., the meaning of the expression $F_i(e_1, \ldots, e_n)$ which results from applying F_i to $e_1, \ldots, e_n$) is equal to the value which the function G_j assigns to the meanings of $e_1, \ldots, e_n$. More formally: let m denote 'the meaning of'. Then:

(30) $m(F_i(e_1, \ldots, e_n)) = G_j(m(e_1), \ldots, m(e_n))$

Consequently, the semantics, too, is a many-sorted algebra. In addition to a family of sets indexed by the types, it includes a number of operators: functions from the Cartesian product of a number of semantic carriers to some semantic carrier.[13]

Hence, thirdly, besides the function σ mapping categories to types, there is a function ρ which associates every n-ary operator F_i in the syntactic algebra with an n-ary operator $G_{\rho(i)}$ in the semantic algebra. And since every syntactic category c is associated with a semantic type $\sigma(c)$, it must be the case that $\omega(\rho(i)) = \langle\langle\sigma(c_1), \ldots, \sigma(c_n)\rangle, \sigma(c_{n+1})\rangle$ whenever $\pi(i) = \langle\langle c_1, \ldots, c_n\rangle, c_{n+1}\rangle$, where π and ω are the type-assigning functions of the syntactic and semantic algebra, respectively.

So, the principle of compositionality dictates the following:

- The syntax is a π-algebra $A = \langle(A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating set $H = (H_s)_{s\in S}$.
- The semantic domain is an ω-algebra $M = \langle(M_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ such that for some $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$: A is (σ, ρ)-interpretable in M.
- Meaning assignment is a (σ, ρ)-homomorphism from $T_{A,H}$, the term algebra of A with respect to H, to M.

The notions '(σ, ρ)-interpretable' and '(σ, ρ)-homomorphism' formalize the explication given above: let $A = \langle(A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a π-algebra, let $B = \langle(B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ be an ω-algebra, and let $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$ be functions. Then:

(31) A is (σ, ρ)-*interpretable in* B iff for all $\gamma \in \Gamma$: if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then $\omega(\rho(\gamma)) = \langle\langle\sigma(s_1), \ldots, \sigma(s_n)\rangle, \sigma(s_{n+1})\rangle$.

[13] We will assume that these are total functions, just as the functions in the syntactic algebra.

A π-algebra A generated by some set H is (σ, ρ)-interpretable in algebra B if and only if $T_{A,H}$ is (σ, ρ)-interpretable in B, for note that A and $T_{A,H}$ invariably have the same set of sorts S, the same set of operator indices Γ, and the same type assigning function π.

Note as well that if A is (σ, ρ)-interpretable in B and B is (σ', ρ')-interpretable in C, then A is $(\sigma \circ \sigma', \rho \circ \rho')$-interpretable in C, where $f \circ g$ denotes the composition of the functions f and g, defined by $f \circ g(x) = g(f(x))$.

Moreover, let π-algebra $A = \langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ be (σ, ρ)-interpretable in ω-algebra $B = \langle (B_t)_{t \in T}, (G_\delta)_{\delta \in \Delta} \rangle$. A (σ, ρ)-homomorphism from A to B is a function from the carriers of A to the carriers of B which respects (i) the sorts, and (ii) the operators ($h[A_s]$ denotes the set $\{h(a) \mid a \in A_s\}$):

(32) $h : \bigcup_{s \in S} A_s \to \bigcup_{t \in T} B_t$ is a (σ, ρ)-*homomorphism* from A to B iff
(i) $h[A_s] \subseteq B_{\sigma(s)}$; and
(ii) if $\pi(\gamma) = \langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$ and $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$, then $h(F_\gamma(a_1, \ldots, a_n)) = G_{\rho(\gamma)}(h(a_1), \ldots, h(a_n))$.

A surjective homomorphism[14] is called an *epimorphism*, and the collection of (σ, ρ)-homomorphisms (epimorphisms) from A to B will be denoted by $\mathrm{Hom}(A, B, \sigma, \rho)$ ($\mathrm{Epi}(A, B, \sigma, \rho)$).

An important property of homomorphisms is that the composition of two homomorphisms h and g is again a homomorphism. Suppose that A, B and C are algebras such that A is (σ, ρ)-interpretable in B and B is (σ', ρ')-interpretable in C, that $h \in \mathrm{Hom}(A, B, \sigma, \rho)$, and that $g \in \mathrm{Hom}(B, C, \sigma', \rho')$. Then $h \circ g \in \mathrm{Hom}(B, C, \sigma \circ \sigma', \rho \circ \rho')$. The proof is straightforward: firstly, recall that A is $(\sigma \circ \sigma', \rho \circ \rho')$-interpretable in C; secondly, since $h[A_s] \subseteq B_{\sigma(s)}$ and $g[B_{\sigma(s)}] \subseteq C_{\sigma'(\sigma(s))}$, necessarily $g[h[A_s]] \subseteq C_{\sigma'(\sigma(s))}$; and thirdly, the following holds for respective operators F_γ, $G_{\rho(\gamma)}$, $H_{\rho'(\rho(\gamma))}$ in A, B, C:

$$
\begin{aligned}
& h \circ g(F_\gamma(a_1, \ldots, a_k)) \\
= \; & g(h(F_\gamma(a_1, \ldots, a_k)) \\
= \; & g(G_{\rho(\gamma)}(h(a_1), \ldots, h(a_k))) \\
= \; & H_{\rho'(\rho(\gamma))}(g(h(a_1)), \ldots, g(h(a_k))) \\
= \; & H_{\rho'(\rho(\gamma))}(h \circ g(a_1), \ldots, (h \circ g(a_k))
\end{aligned}
$$

3 Similarity

The above formalization of the compositionality principle is more or less the same as the one given by Janssen (1986), except for a seemingly minor point

[14] I.e., a homomorphism h such that σ is a *sur*jection and $h[A_s] = B_{\sigma(s)}$ for all $s \in S$.

which will turn out to have rather far-reaching consequences.

Janssen's definition of (σ, ρ)-homomorphism is identical to the one given in (32), but (σ, ρ)-homomorphisms are allowed to exist only between algebras A and B which are '(σ, ρ)-similar'. Here is the definition of that notion (see Janssen (1986), pp. 67–8):

Let $A = \langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ be a π-algebra, let $B = \langle (B_t)_{t \in T}, (G_\delta)_{\delta \in \Delta} \rangle$ be a ω-algebra, and let $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$ be *bijections*. Then:

(33) A and B are (σ, ρ)-*similar* iff for all $\gamma \in \Gamma$: $\omega(\rho(\gamma)) = \langle \langle \sigma(s_1), \ldots, \sigma(s_n) \rangle, \sigma(s_{n+1}) \rangle$ *iff* $\pi(\gamma) = \langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$.

So, (σ, ρ)-similarity differs from (σ, ρ)-interpretability in two respects: the former notion requires (a) that the functions σ and ρ be *bijections*; and (b) that for all $\gamma \in \Gamma$: $\omega(\rho(\gamma)) = \langle \langle \sigma(s_1), \ldots, \sigma(s_n) \rangle, \sigma(s_{n+1}) \rangle$ if *and only if* $\pi(\gamma) = \langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$.

However, given (a) and the 'if'-part of (b), the 'only if'-part of (b) is superfluous,[15] hence requirement (a) is the only real difference between the two notions.

Since bijections are functions, we have that π-algebra A is (σ, ρ)-interpretable in ω-algebra B whenever A and B are (σ, ρ)-similar. The converse does not hold. Therefore, (σ, ρ)-similarity is stronger than (σ, ρ)-interpretability, so that the latter notion is more easily applicable in principle.

Moreover, note that (σ, ρ)-similarity is too strong in view of the explication of the intuitive idea of compositionality we have given in the previous section, but that, on the other hand, (σ, ρ)-interpretability is a notion that *can* be motivated.

Finally, the requirement that the domains of syntax and semantics constitute similar algebras leads to technical complications, as will be shown below.

If we look at the semantic interpretation of the logical language L which was defined in Section 1, both notions turn out to be equally applicable. As we

[15] Assume that σ is an injection (which follows from (a)), and that for all $\gamma \in \Gamma$:

(#) if $\pi(\gamma) = \langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$, then $\omega(\rho(\gamma)) = \langle \langle \sigma(s_1), \ldots, \sigma(s_n) \rangle, \sigma(s_{n+1}) \rangle$.

Consider $\gamma \in \Gamma$ and $\langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$ such that $\pi(\gamma) \neq \langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$. Then $\pi(\gamma) = \langle \langle s'_1, \ldots, s'_m \rangle, s'_{m+1} \rangle$, where $\langle \langle s'_1, \ldots, s'_m \rangle, s'_{m+1} \rangle \neq \langle \langle s_1, \ldots, s_n \rangle, s_{n+1} \rangle$. By (#): $\omega(\rho(\pi)) = \langle \langle \sigma(s'_1), \ldots, \sigma(s'_m) \rangle, \sigma(s'_{m+1}) \rangle$. Moreover, since σ is an injection, we have that $\langle \langle \sigma(s'_1), \ldots, \sigma(s'_m) \rangle, \sigma(s'_{m+1}) \rangle \neq \langle \langle \sigma(s_1), \ldots, \sigma(s_n) \rangle, \sigma(s_{n+1}) \rangle$ and, consequently, that $\omega(\rho(\pi)) \neq \langle \langle \sigma(s_1), \ldots, \sigma(s_n) \rangle, \sigma(s_{n+1}) \rangle$.

have seen, the syntactic algebra B of L is the following ω-algebra:

(34) $\langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$

This syntactic algebra is interpreted in a model m which determines a semantic algebra M:

(35) $\langle (M_\tau)_{\tau\in T}, (G_{m,\gamma})_{\gamma\in\Gamma}\rangle$

This algebra M has the same sets T and Γ of sorts and operator indices as B, and the same type-assigning function ω, viz.:

- T is the smallest set such that $e \in T$, $t \in T$, and if $\sigma \in T$ and $\tau \in T$, then $(\sigma,\tau) \in T$.
- $\Gamma = \{1,2\} \cup (3{:}\tau{:}i)_{\tau\in T, i\in\mathbb{N}} \cup (4{:}\sigma{:}\tau)_{\sigma\in T,\tau\in T} \cup (5{:}\sigma{:}\tau{:}i)_{\sigma\in T,\tau\in T,i\in\mathbb{N}} \cup (6{:}\tau)_{\tau\in T}$.
- $\omega(1) = \langle\langle t\rangle, t\rangle$; $\omega(2) = \langle\langle t,t\rangle, t\rangle$;
  $\omega(3{:}\tau{:}i) = \langle\langle t\rangle, t\rangle$; $\omega(4{:}\sigma{:}\tau) = \langle\langle(\sigma,\tau),\sigma\rangle,\tau\rangle$;
  $\omega(5{:}\sigma{:}\tau{:}i) = \langle\langle\tau\rangle,(\sigma,\tau)\rangle$; $\omega(6{:}\tau) = \langle\langle\tau,\tau\rangle,t\rangle$.

The carriers of M are as follows: for each $\tau \in T$, the carrier M_τ is equal to $D^A_{E,\tau}$, the set of functions from variable assignments to members of the domain $D_{E,\tau}$ (see (6) above). And the operators of M are the functions $(G_{m,\gamma})_{\gamma\in\Gamma}$ which were defined in (10).

The algebras B and M are related by the functions $\sigma : T \to T$ and $\rho : \Gamma \to \Gamma$, defined as $\sigma(\tau) = \tau$ and $\rho(\gamma) = \gamma$, which are, of course, bijections. And since they are identity functions we also have that $\pi(\gamma) = \langle\langle s_1,\ldots,s_n\rangle, s_{n+1}\rangle$ if and only if $\omega(\rho(\gamma)) = \langle\langle\sigma(s_1),\ldots,\sigma(s_n)\rangle,\sigma(s_{n+1})\rangle$. Thus B and M are (σ,ρ)-similar (and B is (σ,ρ)-interpretable in M).

Moreover, the interpretation assignment function in_m defined in (9) is a (σ,ρ)-homomorphism from B to M, since in_m respects the sorts in that for all $\tau \in T$, $in_m[L_\tau] \subseteq M_{\sigma(\tau)} = M_\tau$; and in_m respects the operators: if $\omega(\gamma) = \langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle$ and $a_1 \in L_{\tau_1}, \ldots, a_n \in L_{\tau_n}$, then $in_m(F_\gamma(a_1,\ldots,a_n)) = G_{m,\rho(\gamma)}(in_m(a_1),\ldots,in_m(a_n)) = G_{m,\gamma}(in_m(a_1),\ldots,in_m(a_n))$.

However, if we try to analyze the translation tr (see (18) above) from the syntactic term algebra $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle$ of the English algebra $A = \langle (E_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ with respect to the generating set $K = (K_c)_{c\in C}$ into

the logical algebra $B = \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ in terms of '(σ, ρ)-similarity' and '(σ, ρ)-homomorphism', we run into various problems.

The sorts T, operator indices Γ and type-assigning function ω of B were repeated above. Recall that both A and $T_{A,K}$ have the sorts C, the operator indices Δ and the type-assigning function π, where:

- $C = \{\mathrm{S,T,V,IV,CN,Det,Adv}\}$.
- $\Delta = \{1,2,3,4,5,6\}$.
- $\pi(1) = \langle \langle \mathrm{V} \rangle, \mathrm{IV} \rangle$; $\pi(2) = \langle \langle \mathrm{T}, \mathrm{IV} \rangle, \mathrm{S} \rangle$;
  $\pi(3) = \langle \langle \mathrm{Det}, \mathrm{CN} \rangle, \mathrm{T} \rangle$; $\pi(4) = \langle \langle \mathrm{IV}, \mathrm{Adv} \rangle, \mathrm{IV} \rangle$;
  $\pi(5) = \langle \langle \mathrm{S}, \mathrm{S} \rangle, \mathrm{S} \rangle$; $\pi(6) = \langle \langle \mathrm{IV}, \mathrm{IV} \rangle, \mathrm{IV} \rangle$.

The algebras $T_{A,K}$ and B fail to be (σ, ρ)-similar for the following reasons:

First, there can be no surjection (and hence no bijection) $\sigma : C \to T$, since there are seven sorts in $T_{A,K}$, but infinitely many sorts in B.

Second, the function $\sigma : C \to T$ such that $t \in T_{A,K,c}$ if and only if $tr(t) \in L_{\sigma(c)}$ (see (12)), viz., σ defined as:

$$\sigma(\mathrm{V}) = \sigma(\mathrm{CN}) = \sigma(\mathrm{IV}) = (e,t), \quad \sigma(\mathrm{T}) = ((e,t),t), \quad \sigma(\mathrm{S}) = t,$$
$$\sigma(\mathrm{Det}) = ((e,t),((e,t),t)), \text{ and } \quad \sigma(\mathrm{Adv}) = ((e,t),(e,t)),$$

is not even an injection.

Third, there can be no surjection (and hence no bijection) $\rho : \Delta \to \Gamma$, for there are only six operators in $T_{A,K}$, while B has infinitely many operators.

And fourth, there is no $\rho : \Delta \to \Gamma$ such that if $\pi(\delta) = \langle \langle c_1, \ldots, c_n \rangle, c_{n+1} \rangle$, then $\omega(\rho(\delta)) = \langle \langle \sigma(c_1), \ldots, \sigma(c_n) \rangle, \sigma(c_{n+1}) \rangle$, where σ is as above. We have:

$$\pi(1) = \langle \langle \mathrm{V} \rangle, \mathrm{IV} \rangle,$$
$$\pi(4) = \langle \langle \mathrm{IV}, \mathrm{Adv} \rangle, \mathrm{IV} \rangle, \text{ and}$$
$$\pi(6) = \langle \langle \mathrm{IV}, \mathrm{IV} \rangle, \mathrm{IV} \rangle;$$

but there is no $\gamma \in \Gamma$ (and hence no $\rho(\delta) \in \Gamma$) such that:

$$\omega(\gamma) = \langle \langle \sigma(\mathrm{V}) \rangle, \sigma(\mathrm{IV}) \rangle = \langle \langle (e,t) \rangle, (e,t) \rangle,$$
$$\omega(\gamma) = \langle \langle \sigma(\mathrm{IV}), \sigma(\mathrm{Adv}) \rangle, \sigma(\mathrm{IV}) \rangle = \langle \langle (e,t), ((e,t),(e,t)) \rangle, (e,t) \rangle, \text{ or}$$
$$\omega(\gamma) = \langle \langle \sigma(\mathrm{IV}), \sigma(\mathrm{IV}) \rangle, \sigma(\mathrm{IV}) \rangle = \langle \langle (e,t), (e,t) \rangle, (e,t) \rangle.$$

Consequently, there cannot be a (σ, ρ)-homomorphism from $T_{A,K}$ to A.

On the other hand, observe that $T_{A,K}$ is 'almost' (σ, ρ)-interpretable in B. The operators H_2^T, H_3^T and H_5^T in $T_{A,K}$ correspond to operators which are already present in B, viz., $F_{4:(e,t):t}$, $F_{4:(e,t):((e,t),t)}$ and F_2, respectively. Therefore, for $\sigma : C \to T$ defined in (12) and for any function $\rho : \Delta \to \Gamma'$ such that $\{x, y, z\} \cup \Gamma \subseteq \Gamma'$ and

$$\begin{aligned}
&\rho(1) = x,\\
&\rho(2) = 4{:}(e,t){:}t,\\
&\rho(3) = 4{:}(e,t){:}((e,t),t),\\
&\rho(4) = y,\\
&\rho(5) = 2, \text{ and}\\
&\rho(6) = z,
\end{aligned}$$

it holds that the π-algebra $T_{A,K}$ is (σ, ρ)-interpretable in ω'-algebra $B' = \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma'} \rangle$, provided that $\omega \subseteq \omega'$ and that

$$\begin{aligned}
&\omega'(x) = \langle\langle (e,t)\rangle, (e,t)\rangle,\\
&\omega'(y) = \langle\langle (e,t), ((e,t),(e,t))\rangle, (e,t)\rangle, \text{ and}\\
&\omega'(z) = \langle\langle (e,t), (e,t)\rangle, (e,t)\rangle.
\end{aligned}$$

And if, moreover, the additional operators F_x, F_y and F_z in B' are defined as in (36), (37) and (38) below, then the translation function tr defined in (18) is a (σ, ρ)-homomorphism from $T_{A,K}$ to B', since tr respects the sorts: for every $c \in C$, $tr[T_{A,K,c}] \subseteq L_{\sigma(c)}$; and tr respects the operators: if $\pi(\delta) = \langle\langle c_1, \ldots, c_n\rangle, c\rangle$ and $t_1 \in T_{A,K,c_1}, \ldots, t_n \in T_{A,K,c_n}$, then $tr(H_\delta^T(t_1, \ldots, t_n)) = F_{\rho(\delta)}(tr(t_1), \ldots, tr(t_n))$.[16]

(36) $F_x(\alpha) = \alpha$

(37) $F_y(\alpha, \beta) = F_{4:(e,t):(e,t)}(\beta, \alpha)$

(38) $F_z(\alpha, \beta) = F_{5:e:t:0}(F_2(F_{4:e:t}(\alpha, v_{0,e}), F_{4:e:t}(\beta, v_{0,e})))$

Note that the 'new' operators F_x, F_y and F_z are either trivial (F_x), or definable in terms of operators already present (F_y and F_z). We will show

[16] Recall (from (4)) that:

- $F_{4:(e,t):(e,t)} : L_{((e,t),(e,t))} \times L_{(e,t)} \to L_{(e,t)}$, and $F_{4:(e,t):(e,t)}(\alpha, \beta) = [[\alpha](\beta)]$;
- $F_{5:e:t:0} : L_t \to L_{(e,t)}$, and $F_{5:e:t:0}(\alpha) = \lambda v_{0,e}\ \alpha$;
- $F_2 : L_t \times L_t \to L_t$, and $F_2(\varphi, \psi) = [\varphi \wedge \psi]$;
- $F_{4:e:t} : L_{(e,t)} \times L_e \to L_t$, and $F_{4:e:t}(\alpha, \beta) = [[\alpha](\beta)]$;

Thus $F_z(\alpha, \beta) = \lambda v_{0,e}\ [[[\alpha](v_{0,e})] \wedge [[\beta](v_{0,e})]] = \lambda x\ [[[\alpha](x)] \wedge [[\beta](x)]]$ and $F_y(\alpha, \beta) = [[\beta](\alpha)]$ (cf. (19)).

below (see Section 3.1 and Section 4) that the addition of such operators is always unproblematic.

In order to bridge gaps of dissimilarity between algebras, such as the one between the term algebra $T_{A,K}$ for the fragment of English and the logical algebra B, Janssen invokes the notion of a 'safe deriver'.

This notion is introduced in the course of giving a definition of a Montague grammar, which, in its most simple form, consists of a many-sorted algebra and a homomorphic interpretation. However,

> one always uses, in practice, some formal (logical) language as auxiliary language, and the language of which one wishes to define the meanings is translated into this formal language. Thus the meaning assignment is performed indirectly. The aspect of translating into an auxiliary language is, in my opinion, unavoidable for practical reasons, and I therefore wish to incorporate this aspect in the definition of a Montague grammar. (Janssen (1986), Part 1, p. 81)

This definition is given in (39):

(39) A *Montague grammar* consists of:
- (*a*) an algebra A (the *syntactic algebra*) generated by K;
- (*b*) an algebra B (the *logical algebra*);
- (*c*) an algebra M similar to B (the *semantic algebra*);
- (*d*) a homomorphism $h \in \mathrm{Hom}(B, M)$ (the *interpretation of* B);
- (*e*) an algebra $D(B)$ similar to A, where D is a *safe deriver*;
- (*f*) a homomorphism $tr \in \mathrm{Hom}(T_{A,K}, D(B))$ (the *translation*).

The situation can be sketched as follows (cf. Janssen (1986), pp. 75 and 82):

(40)
$$\begin{array}{ccc} & & T_{A,K} \\ & & \downarrow tr \\ B & \Longrightarrow & D(B) \\ \downarrow h & & \downarrow h \\ M & \Longrightarrow & M' \end{array}$$

In general, a *deriver* D is a function from algebras to algebras: 'a method to obtain new algebras from old ones' (Janssen (1986), p. 76), and:

(41) A deriver D is *safe* for algebra P iff for all algebras Q and all $h \in \mathrm{Epi}(P, Q)$ there is a unique algebra Q' such that for the restriction h' of h to $D(P)$ it holds that $h' \in \mathrm{Epi}(D(P), Q')$.

Janssen offers no arguments why this should define the safeness of a deriver. The only motivation given is the following:

> The requirement that h' is an epimorphism is important. If we would not require this, then Q' would in most cases not be unique. An extreme example arises when $D(P)$ is an empty algebra. Then there are infinitely many algebras Q' such that $h' \in \mathrm{Hom}(D(P), Q')$, but only one such that $h' \in \mathrm{Epi}(D(P), Q')$. (Janssen (1986), Part 1, p. 76)

At first sight, one would think that the algebras B, $D(B)$, M and M' in picture (40) instantiate the algebras P, $D(P)$, Q and Q' mentioned in definition (41), respectively. But this is problematic. It cannot be assumed in general that the homomorphism h from the logical algebra B to the semantic algebra M is an epimorphism.[17]

Therefore, we will instead assume that it is not the semantic algebra $M = \langle (M_{\sigma(\tau)})_{\tau \in T}, (G_{\rho(\gamma)})_{\gamma \in \Gamma} \rangle$ itself which plays the role of algebra Q in (40), but that it is the subalgebra N of M which consists of the interpretations that the (σ, ρ)-homomorphism h assigns to expressions of the logical algebra $B = \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma} \rangle$. That is: $N = \langle (h[L_\tau]_{\sigma(\tau)})_{\tau \in T}, (G'_{\rho(\gamma)})_{\gamma \in \Gamma} \rangle$, where $h[L_\tau]$ denotes the set $\{h(a) \mid a \in L_\tau\}$, and G'_γ denotes the restriction of G_γ to $(h[L_\tau]_{\sigma(\tau)})_{\tau \in T}$. Of course, the function h cannot fail to be an epimorphism from B to N. The unique algebra M' in (40) which is determined by the safe derivation of $D(B)$ from B, then, consists of the interpretations of expressions of $D(B)$. These interpretations are assigned by the restriction h' of h to $D(B)$, and not by h itself, as is suggested in (40). Finally, the interpretation of terms in $T_{A,K}$ is taken care of by $tr \circ h'$, the composition of the homomorphism tr and the epimorphism h', which is itself a

[17] Thus in the case we studied, that of the algebra (34) $= \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ of extensional typed logic with infinitely many variables of each type, every expression α of type τ is homomorphically assigned an interpretation $in_m(\alpha)$ in the semantic algebra (35) $= \langle (M_\tau)_{\tau \in T}, (G_{m,\gamma})_{\gamma \in \Gamma} \rangle$, where $in_m(\alpha)$ is a member of the carrier $M_\tau = D^A_{E,\tau}$, the set of functions from variable assignments to objects in $D_{E,\tau}$. It is easily shown that in_m cannot be an epimorphism. Note, for instance, that no expression φ of type t – the domain $D_{E,t}$ of which is the set $\{0, 1\}$ – can denote the characteristic function of a singleton set of variable assignments. This is impossible because the truth value denoted by φ under an variable assignment a is only dependent on the (free) variables occurring in φ, of which there are only finitely many. Hence, if a is an arbitrary variable assignment and $v_{i,t}$ is some variable of type t which does not occur in φ, then $\langle a[v_{i,t}/1], 1 \rangle \in \varphi$ iff $\langle a[v_{i,t}/0], 1 \rangle \in \varphi$. So, necessarily the interpretation homomorphism h is not surjective.

homomorphism from $T_{A,K}$ to M' (cf. the comments on (32) above).[18]

In sum, this interpretation entails the following revision of the picture drawn in (39) and (40):

1. M is not some algebra similar to B, but the image of B under h.
2. The leftmost homomorphism h in (40) is an epimorphism.
3. The rightmost h in (40) must be replaced by h', the restriction of h to $D(B)$.

Janssen's deriver D is the composition of four basic derivers, viz., AddOp, AddSorts, DelOp and DelSorts.[19] Let

$$\begin{aligned} B' &= \text{AddOp}(B); \\ B'' &= \text{AddSorts}(B'); \\ B''' &= \text{DelOp}(B''), \text{ and} \\ B'''' &= \text{DelSorts}(B'''). \end{aligned}$$

Then $D(B) = B'''' = \text{DelSorts}(\text{DelOp}(\text{AddSorts}(\text{AddOp}(B))))$. Moreover, let h be an epimorphism from B to N, and let h', h'', h''' and h'''' be the restriction of h, h', h'', and h''' to B', B'', B''' and B'''', respectively.

(42)
$$\begin{array}{ccccccccc} B & \Longrightarrow & B' & \Longrightarrow & B'' & \Longrightarrow & B''' & \Longrightarrow & B'''' \\ \downarrow h & & \downarrow h' & & \downarrow h'' & & \downarrow h''' & & \downarrow h'''' \\ N & \Longrightarrow & N' & \Longrightarrow & N'' & \Longrightarrow & N''' & \Longrightarrow & N'''' \end{array}$$

If the basic derivers AddOp, AddSorts, DelOp and DelSorts are all safe, then the algebras N', N'', N''' and N'''' are unique.[20]

Now, taking (*i*) the logical ω-algebra (34) (i.e., $B = \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma} \rangle$); (*ii*) the subalgebra $N = \langle (in_m[L_\tau]_\tau)_{\tau \in T}, (G'_{m,\gamma})_{\gamma \in \Gamma} \rangle$ of the semantic ω-algebra (35) (i.e., $M = \langle (M_\tau)_{\tau \in T}, (G_{m,\gamma})_{\gamma \in \Gamma} \rangle$); and (*iii*) the (σ, ρ)-epimorphism[21] in_m from B to N as our starting point, we will give a description of

[18]Due to the transitivity of similarity (if A and B are (σ, ρ)-similar and B and C are (σ', ρ')-similar, then A and C are $(\sigma \circ \sigma', \rho \circ \rho')$-similar), the algebras $T_{A,K}$ to M' are similar.

[19]In fact, the deriver DelSorts replaces the deriver SubAlg actually proposed by Janssen. See footnote 34 below for motivation.

[20]Strictly speaking, they are unique up to isomorphism, cf. footnote 29 below.

[21]Recall that in_m is a (σ, ρ)-homomorphism from B to M for the functions $\sigma : T \to T$ and $\rho : \Gamma \to \Gamma$ such that $\sigma(\tau) = \tau$ and $\rho(\gamma) = \gamma$. Hence in_m is a (σ, ρ)-epimorphism from B to N for the same σ and ρ.

the four basic derivers, and show how they can be used to 'safely derive' an algebra $D(B)$ which (a) is similar to the term algebra $T_{A,K}$ of the English algebra A generated by K, and (b) determines a unique algebra N'''' such that for the restriction in'_m of in_m to $D(B)$ it holds that in'_m is an epimorphism from $D(B)$ to N''''.

3.1 AddOp

The first and 'most important' (Janssen (1986), p. 82) deriver is AddOp. An application of AddOp to a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ has the form $\mathrm{AddOp}[(F_\gamma)_{\gamma\in\Gamma'}](A)$, where ($i$) $\Gamma \cap \Gamma' = \emptyset$; and ($ii$) for every $\gamma \in \Gamma'$ there is one sequence of sorts $\vec{\gamma} = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$ (where $n \in \mathbb{N}^+$) such that $F_\gamma : A_{s_1} \times \ldots \times A_{s_n} \to A_{s_{n+1}}$. The resulting algebra $\mathrm{AddOp}[(F_\gamma)_{\gamma\in\Gamma'}](A)$ is the algebra $\langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma\cup\Gamma'}\rangle$ of signature $\pi' = \pi \cup \{\langle\gamma, \vec{\gamma}\rangle \mid \gamma \in \Gamma'\}$.

It will be shown below that if the set $(F_\gamma)_{\gamma\in\Gamma'}$ of added operators only contains polynomial operators over A, then $\mathrm{AddOp}[(F_\gamma)_{\gamma\in\Gamma'}]$ is safe.

The set of polynomial operators over an algebra $A = \langle (A_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ consists, roughly speaking, of a number of elementary operators (projection functions and constant functions) plus operators which are definable in terms of these elementary operators and the operators in $(F_\gamma)_{\gamma\in\Gamma}$. We will now give an exact definition in terms of so-called polynomial symbols.[22]

Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a many-sorted algebra of signature π. On the basis of A, we first define a couple of sets of auxiliary symbols. Let $s \in S$ and let, for $n \in \mathbb{N}^+$, the sequence $\langle\langle s_1, \ldots, s_n\rangle, s\rangle \in S^n \times S$. Then:

(43) $\mathrm{VAR}_s = \{\xi^i_s \mid i \in \mathbb{N}^+\}$,
$\mathrm{CON}^A_s = \{a_s \mid a \in A_s\}$, and
$\mathrm{OP}^A_{\langle\langle s_1,\ldots,s_n\rangle,s\rangle} = \{\underline{F}_\gamma \mid F_\gamma \in (F_\gamma)_{\gamma\in\Gamma}$ and $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s\rangle\}$.

VAR_s is the set of *polynomial variables* of sort s, CON^A_s is the set of *polynomial constants* over A of sort s, and $\mathrm{OP}^A_{\langle\langle s_1,\ldots,s_n\rangle,s\rangle}$ is the set of *polynomial operator symbols* over A of type $\langle\langle s_1, \ldots, s_n\rangle, s\rangle$.

For $n \in \mathbb{N}^+$ and $\langle\vec{s}, s\rangle \in S^n \times S$, the set $\mathrm{POL}^A_{\langle\vec{s},s\rangle}$ of *polynomial symbols*

[22] Cf. Janssen (1986), pp. 56–61. See Montague (1970b), p. 224, for a direct (one-sorted) definition. In the many-sorted context, the mediation of polynomial symbols leads to a more neatly arranged bureaucracy.

over A of type $\langle \vec{s}, s\rangle$ is defined as the smallest set such that:

(44) (*i*) if $\vec{s} = \langle s_1, \ldots, s_{i-1}, s, s_{i+1} \ldots, s_n\rangle$, then $\lfloor \xi^i_s \rfloor \in \mathrm{POL}^A_{\langle \vec{s}, s\rangle}$;
(*ii*) if $a_s \in \mathrm{CON}^A_s$, then $\lfloor a_s \rfloor \in \mathrm{POL}^A_{\langle \vec{s}, s\rangle}$; and
(*iii*) if $p_1 \in \mathrm{POL}^A_{\langle \vec{s}, s_1\rangle}, \ldots, p_k \in \mathrm{POL}^A_{\langle \vec{s}, s_k\rangle}$ and $\underline{F}_\gamma \in \mathrm{OP}^A_{\langle\langle s_1, \ldots, s_k\rangle, s\rangle}$, then $\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor \in \mathrm{POL}^A_{\langle \vec{s}, s\rangle}$.

(45) $\mathrm{POL}^A = \bigcup\{\, \mathrm{POL}^A_{\langle \vec{s}, s\rangle} \mid \langle \vec{s}, s\rangle \in S^n \times S \text{ and } n \in \mathbb{N}^+\}$.

A polynomial symbol $p \in \mathrm{POL}^A_{\langle \vec{s}, s\rangle}$ uniquely determines a *polynomial operator* $p^\#$ of type $\langle \vec{s}, s\rangle$. Let $\vec{s}$ be the sequence $\langle s_1, \ldots, s_n\rangle$:

(46) If $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$, then $p^\#(a_1, \ldots, a_n)$ is defined by
(*i*) $\lfloor \xi^i_s \rfloor^\#(a_1, \ldots, a_n) = a_i$,
(*ii*) $\lfloor c_s \rfloor^\#(a_1, \ldots, a_n) = c$, and
(*iii*) $\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor^\#(a_1, \ldots, a_n) = F_\gamma(p_1^\#(a_1, \ldots, a_n), \ldots, p_k^\#(a_1, \ldots, a_n))$.

Thus (*i*) polynomial symbols $\lfloor \xi^i_s \rfloor$ of type $\langle\langle s_1, \ldots s_n\rangle, s\rangle$ determine a *projection function* $\lfloor \xi^i_s \rfloor^\# : A_{s_1} \times \ldots \times A_{s_n} \to A_s$ which yields its i-th argument as a result; (*ii*) polynomial symbols $\lfloor c_s \rfloor$ of type $\langle\langle s_1, \ldots s_n\rangle, s\rangle$ determine a *constant function* $\lfloor c_s \rfloor^\# : A_{s_1} \times \ldots \times A_{s_n} \to A_s$ which yields $c \in A_s$ as a result; and (*iii*) polynomial symbols $\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor$ of type $\langle\langle s_1, \ldots s_n\rangle, s\rangle$ determine a function $\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor^\#$ which is a *composition* of F_γ and the functions $p_1^\#, \ldots, p_k^\#$ determined by the respective polynomial symbols $p_1, \ldots, p_k$.[23]

Let $(P_\delta)_{\delta \in \Delta}$ be a set of polynomial symbols over $A = \langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma}\rangle$. Projection functions yield one of their arguments as a result, constant functions yield a member of a carrier as a result, and the collection $(A_s)_{s \in S}$ is closed under the operations $F_\gamma \in (F_\gamma)_{\gamma \in \Gamma}$, hence the addition of $(P_\delta^\#)_{\delta \in \Delta}$ yields an algebra $\langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma} \cup (P_\delta^\#)_{\delta \in \Delta}\rangle$ with the same carriers as A.

[23] Note that the operators F_γ in a π-algebra $\langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma}\rangle$ are themselves expressed by a polynomial symbol: if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s\rangle$, then $F_\gamma = \lfloor \underline{F}_\gamma \lfloor \xi^1_{s_1} \rfloor \ldots \lfloor \xi^n_{s_n} \rfloor \rfloor^\#$, where $\lfloor \xi^1_{s_1} \rfloor, \ldots, \lfloor \xi^n_{s_n} \rfloor$ have the respective types $\langle\langle s_1, \ldots, s_n\rangle, s_1\rangle, \ldots, \langle\langle s_1, \ldots, s_n\rangle, s_n\rangle$.

Incidentally, the members of a term algebra $T_{A,H}$ of $A = \langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma}\rangle$ with respect to generating set $H = (H_s)_{s \in S}$ are the polynomial symbols which (*i*) are built up from polynomial constants h_s corresponding to members h of some H_s and operator symbols $\underline{F}_\gamma$ corresponding to some F_γ, and (*ii*) contain no polynomial variables.

AddOp can be employed to derive an algebra B' from the logical algebra B which is 'more similar' to the English term algebra $T_{A,K}$. We add three polynomial operators, $F_x : L_{(e,t)} \rightarrow L_{(e,t)}$, $F_y : L_{(e,t)} \times L_{((e,t),(e,t))} \rightarrow L_{(e,t)}$ and $F_z : L_{(e,t)} \times L_{(e,t)} \rightarrow L_{(e,t)}$, to algebra B, which will correspond to the respective operators H_1^T, H_4^T and H_6^T in $T_{A,K}$. These additional operators are determined by the polynomial symbols

- $P_x \in \mathrm{POL}^B_{\langle\langle(e,t)\rangle,(e,t)\rangle}$,
- $P_y \in \mathrm{POL}^B_{\langle\langle(e,t),((e,t),(e,t))\rangle,(e,t)\rangle}$, and
- $P_z \in \mathrm{POL}^B_{\langle\langle(e,t),(e,t)\rangle,(e,t)\rangle}$,

in that $F_x = P_x^{\#}$, $F_y = P_y^{\#}$, and $F_z = P_z^{\#}$, where:

- $P_x = \lfloor \xi^1_{(e,t)} \rfloor$,[24]
- $P_y = \lfloor \underline{F}_{4:(e,t):(e,t)} \lfloor \xi^2_{((e,t),(e,t))} \rfloor \lfloor \xi^1_{(e,t)} \rfloor \rfloor$,[25] and
- $P_z = \lfloor \underline{F}_{5:e:t:0} \lfloor \underline{F}_2 \lfloor \underline{F}_{4:e:t} \lfloor \xi^1_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor \lfloor \underline{F}_{4:e:t} \lfloor \xi^2_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor \rfloor \rfloor$.[26]

The polynomial symbols P_x, P_y and P_z determine the operators which were defined in (36), (37) and (38), respectively, viz.:

- $P_x^{\#}(\alpha) = F_x(\alpha) = \alpha$,
- $P_y^{\#}(\alpha, \beta) = F_y(\alpha, \beta) = [[\beta](\alpha)]$, and
- $P_z^{\#}(\alpha, \beta) = F_z(\alpha, \beta) = \lambda v_{0,e}\ [[[\alpha](v_{0,e})] \wedge [[\beta](v_{0,e})]]$.

[24] The polynomial *variable* $\xi^1_{(e,t)}$ in P_x is of *sort* (e,t), but the polynomial *symbol* $\lfloor \xi^1_{(e,t)} \rfloor$ is of *type* $\langle\langle(e,t)\rangle,(e,t)\rangle$.

[25] The symbols $\lfloor \xi^1_{(e,t)} \rfloor$ and $\lfloor \xi^2_{((e,t),(e,t))} \rfloor$ in P_y are of type $\langle\langle(e,t),((e,t),(e,t))\rangle,(e,t)\rangle$ and $\langle\langle(e,t),((e,t),(e,t))\rangle,((e,t),(e,t))\rangle$, respectively, so that P_y as a whole is of type $\langle\langle(e,t),((e,t),(e,t))\rangle,(e,t)\rangle$.

[26] In P_z, the *logical variable* $v_{0,e}$ figures in the *polynomial constant* $v_{0,e,e}$ of sort e, which is responsible for the polynomial symbol $\lfloor v_{0,e,e} \rfloor$ of type $\langle\langle(e,t),(e,t)\rangle,e\rangle$. Both $\xi^1_{(e,t)}$ and $\xi^2_{(e,t)}$ are of type $\langle\langle(e,t),(e,t)\rangle,(e,t)\rangle$, so that $\lfloor \underline{F}_{4:e:t} \lfloor \xi^1_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor$ and $\lfloor \underline{F}_{4:e:t} \lfloor \xi^2_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor$ have type $\langle\langle(e,t),(e,t)\rangle,t\rangle$. As a consequence, the latter type is also the type of $\lfloor \underline{F}_2 \lfloor \underline{F}_{4:e:t} \lfloor \xi^1_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor \lfloor \underline{F}_{4:e:t} \lfloor \xi^2_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor \rfloor$, and P_z as a whole is of type $\langle\langle(e,t),(e,t)\rangle,(e,t)\rangle$.

Observe that $B' = \mathrm{AddOp}[(F_\gamma)_{\gamma\in\{x,y,z\}}](B) = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma\cup\{x,y,z\}})\rangle$ is an algebra of signature ω', where

$$\begin{aligned}\omega' = \omega \cup \{ &\langle x, \langle\langle (e,t)\rangle, (e,t)\rangle,\\ &\langle y, \langle\langle (e,t), ((e,t),(e,t))\rangle, (e,t)\rangle,\\ &\langle z, \langle\langle (e,t), (e,t)\rangle, (e,t)\rangle \}.\end{aligned}$$

Next, let Γ, Γ', Δ and Δ' be sets of operator indices such that $\Gamma\cap\Gamma' = \emptyset$ and $\Delta\cap\Delta' = \emptyset$, and let $\rho' : \Gamma\cup\Gamma' \to \Delta\cup\Delta'$ be a bijection such that $\rho \subseteq \rho'$. The 'safeness' of adding a set $(P^{\#}_\gamma)_{\gamma\in\Gamma'}$ of polynomial operators over a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ to A amounts to the following.

Suppose that A and ω-algebra $B = \langle (B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ are (σ,ρ)-similar, that $h \in \mathrm{Epi}(A, B, \sigma, \rho)$, and that $A' = \mathrm{AddOp}[(P^{\#}_\gamma)_{\gamma\in\Gamma'}](A)$. Then there is a unique algebra B' such that $h \in \mathrm{Epi}(A', B', \sigma, \rho')$,[27] viz., $B' =$

$$\mathrm{AddOp}[(f(P_\gamma)^{\#}_{\rho'(\gamma)})_{\gamma\in\Gamma'}](B).$$

The function f assigns a polynomial symbol in $\mathrm{POL}^{B}_{\langle\langle\sigma(s_1),\ldots\sigma(s_n)\rangle,\sigma(s)\rangle}$ to each polynomial symbol in $\mathrm{POL}^{A}_{\langle\langle s_1,\ldots s_n\rangle,s\rangle}$ and is defined as follows:

(a) $f(\lfloor\xi^i_s\rfloor) = \lfloor\xi^i_{\sigma(s)}\rfloor$

(b) $f(\lfloor c_s\rfloor) = \lfloor h(c)_{\sigma(s)}\rfloor$

(c) $f(\lfloor\underline{F}_\gamma\, p_1 \ldots p_k\rfloor) = \lfloor\underline{G}_{\rho'(\gamma)}\, f(p_1)\ldots f(p_k)\rfloor$

First, note that $h \in \mathrm{Epi}(A', B', \sigma, \rho')$: since A' and B' have the same carriers as A and B, respectively, we know that h is a surjection which respects the sorts. Moreover, the function h respects not only the 'old' operators in $(F_\gamma)_{\gamma\in\Gamma}$, but also the 'new' ones in $(P^{\#}_\gamma)_{\gamma\in\Gamma'}$, because it holds that $h(P^{\#}_\gamma(a_1,\ldots,a_n)) = (f(P_\gamma))^{\#}(h(a_1),\ldots,h(a_n))$. This can be shown by induction on $c(P)$, the complexity[28] of $P \in \mathrm{POL}^A$:

(a) $h(\lfloor\xi^i_s\rfloor^{\#}(a_1,\ldots,a_n)) = h(a_i) = \lfloor\xi^i_{\sigma(s)}\rfloor^{\#}(h(a_1),\ldots,h(a_n)) = (f(\lfloor\xi^i_s\rfloor))^{\#}(h(a_1),\ldots,h(a_n))$;

[27] We have seen that the addition of polynomial operators to an algebra does not affect the carriers. Hence the restriction h' of h to A' is the same function as h.

[28] That is: $c(\lfloor\xi^i_s\rfloor) = c(\lfloor c_s\rfloor) = 0$, and $c(\lfloor\underline{F}_\gamma\, \lfloor p_1\rfloor\ldots\lfloor p_n\rfloor\rfloor) = \max(c(p_1),\ldots,c(p_n)) + 1$.

(*b*) $h(\lfloor c_s \rfloor^{\#}(a_1, \ldots, a_n)) = h(c) = \lfloor h(c)_{\sigma(s)} \rfloor^{\#}(h(a_1), \ldots, h(a_n)) = (f(\lfloor c_s \rfloor))^{\#}(h(a_1), \ldots, h(a_n))$; and

(*c*) $h(\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor^{\#}(a_1, \ldots, a_n)) =$
$h(F_\gamma(p_1^{\#}(a_1, \ldots, a_n), \ldots, p_k^{\#}(a_1, \ldots, a_n))) =$
$G_{\rho(\gamma)}(h(p_1^{\#}(a_1, \ldots, a_n)), \ldots, h(p_k^{\#}(a_1, \ldots, a_n))) =$
$G_{\rho(\gamma)}((f(p_1))^{\#}(h(a_1), \ldots, h(a_n)), \ldots, (f(p_k))^{\#}(h(a_1), \ldots, h(a_n))) =$
$\lfloor \underline{G}_{\rho(\gamma)}\, f(p_1) \ldots f(p_k) \rfloor^{\#}(h(a_1), \ldots, h(a_n)) =$
$(f(\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor))^{\#}(h(a_1), \ldots, h(a_n))$.

Second, B' is the only algebra such that $h \in \mathrm{Epi}(A', B', \sigma, \rho')$. Suppose that $h \in \mathrm{Epi}(A', B'', \sigma, \rho')$. Then A and B'' are (σ, ρ')-similar, so B'' has the sorts $\{\sigma(s) \mid s \in S\}$, the operator indices $\{\rho'(\gamma) \mid \gamma \in \Gamma \cup \Gamma'\}$ and the type-assigning function $\omega \cup \{\langle \rho'(\gamma), \langle \langle \sigma(s_1), \ldots, \sigma(s_n) \rangle, \sigma(s) \rangle \mid P_\gamma \in \mathrm{POL}^{A}_{\langle \langle s_1, \ldots, s_n \rangle, s \rangle}\}$. Moreover: $B''_{\sigma(s)} = h[A'_s]$ for all $\sigma(s)$ such that $s \in S$, and $G''_{\rho'(\gamma)}(b_1, \ldots, b_n) = G''_{\rho'(\gamma)}(h(a_1), \ldots, h(a_n)) = h(F_\gamma(a_1, \ldots, a_n))$ for all operators $G''_{\rho'(\gamma)}$ in B''.[29]

As regards our example, note that for the algebras $B = \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma} \rangle$, $N = \langle (in_m[L_\tau]_\tau)_{\tau \in T}, (G'_{m,\gamma})_{\gamma \in \Gamma} \rangle$ and the epimorphism in_m from B to N, the safeness of AddOp guarantees that there is a unique algebra N' such that $in_m \in \mathrm{Epi}(\mathrm{AddOp}[(F_\gamma)_{\gamma \in \{x,y,z\}}](B), N')$. This algebra N' results from N by adding

- $G'_{m,x} : in_m[L_{(e,t)}] \to in_m[L_{(e,t)}]$,
- $G'_{m,y} : in_m[L_{(e,t)}] \times in_m[L_{((e,t),(e,t))}] \to in_m[L_{(e,t)}]$ and
- $G'_{m,z} : in_m[L_{(e,t)}] \times in_m[L_{(e,t)}] \to in_m[L_{(e,t)}]$

to the operators of N. These additional operators, which correspond to the operators F_x $(= P_x^{\#})$, F_y $(= P_y^{\#})$ and F_z $(= P_z^{\#})$ in $\mathrm{AddOp}[(F_\gamma)_{\gamma \in \{x,y,z\}}](B)$, are determined by the polynomial symbols

[29] In fact, the uniqueness of B' is a consequence of our fixing specific bijections σ and ρ', which determines the sorts, operator indices and type-assigning function of B'. If this is not done, then B' is only unique 'up to isomorphism'. A π-algebra $A = \langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ and a ω-algebra $B = \langle (B_t)_{t \in T}, (G_\delta)_{\delta \in \Delta} \rangle$ are *identical up to isomorphism* if and only if they have the 'same' carriers and operators. I.e., there are bijections $\varphi : S \to T$ and $\psi : \Gamma \to \Delta$ such that for all $s \in S$ and $\gamma \in \Gamma$: (*i*) $A_s = B_{\varphi(s)}$; (*ii*) $\pi(\gamma) = \langle \langle s_1, \ldots, s_n \rangle, s \rangle$ iff $\omega(\psi(\gamma)) = \langle \langle \varphi(s_1), \ldots, \varphi(s_n) \rangle, \varphi(s) \rangle$; and (*iii*) $\langle \langle a_1, \ldots, a_n \rangle, a \rangle \in F_\gamma$ iff $\langle \langle a_1, \ldots, a_n \rangle, a \rangle \in G_{\psi(\gamma)}$. This proviso of isomorphism applies as well to the derivers to be discussed below. For convenience, we will assume there that epimorphisms connect (σ, ρ)-similar algebras A and B for *identity functions* σ and ρ.

- $f(P_x) \in \mathrm{POL}^N_{\langle\langle(e,t)\rangle,(e,t)\rangle}$,
- $f(P_y) \in \mathrm{POL}^N_{\langle\langle(e,t),((e,t),(e,t))\rangle,(e,t)\rangle}$, and
- $f(P_z) \in \mathrm{POL}^N_{\langle\langle(e,t),(e,t)\rangle,(e,t)\rangle}$,

in that $G'_{m,x} = (f(P_x))^{\#}$, $G'_{m,y} = (f(P_y))^{\#}$, and $G'_{m,z} = (f(P_z))^{\#}$, where:

- $f(P_x) = f(\lfloor \xi^1_{(e,t)} \rfloor) = \lfloor \xi^1_{(e,t)} \rfloor$,
- $f(P_y) = f(\lfloor \underline{F}_{4:(e,t):(e,t)} \lfloor \xi^2_{((e,t),(e,t))} \rfloor \lfloor \xi^1_{(e,t)} \rfloor \rfloor) =$
  $\lfloor \underline{G'_m}_{4:(e,t):(e,t)} \lfloor \xi^2_{((e,t),(e,t))} \rfloor \lfloor \xi^1_{(e,t)} \rfloor \rfloor$, and
- $f(P_z) = f(\lfloor \underline{F}_{5:e:t:0} \lfloor \underline{F}_2 \lfloor \underline{F}_{4:e:t} \lfloor \xi^1_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor \lfloor \underline{F}_{4:e:t} \lfloor \xi^2_{(e,t)} \rfloor \lfloor v_{0,e,e} \rfloor \rfloor \rfloor \rfloor) =$
  $\lfloor \underline{G'_m}_{5:e:t:0} \lfloor \underline{G'_m}_2 \lfloor \underline{G'_m}_{4:e:t} \lfloor \xi^1_{(e,t)} \rfloor \lfloor in_m(v_{0,e})_e \rfloor \rfloor$
  $\lfloor \underline{G'_m}_{4:e:t} \lfloor \xi^2_{(e,t)} \rfloor \lfloor in_m(v_{0,e})_e \rfloor \rfloor \rfloor \rfloor$.

The operators determined by the respective polynomial symbols $f(P_x)$, $f(P_y)$ and $f(P_z)$ are defined by:

- $G'_{m,x}(\alpha) = \alpha$,
- $G'_{m,y}(\alpha, \beta) =$
  $G'_{m,4:(e,t):(e,t)}(\beta, \alpha) =$
  $\{\langle a, f(d)\rangle \mid \langle a, f\rangle \in \beta \text{ and } \langle a, d\rangle \in \alpha\}$, and
- $G'_{m,z}(\alpha, \beta) =$
  $G'_{m,5:e:t:0}(G'_{m,2}(G'_{m,4:e:t}(\alpha, in_m(v_{0,e})), G'_{m,4:e:t}(\alpha, in_m(v_{0,e})))) =$
  $\{\langle a, h\rangle \mid h(d) = 1 \text{ iff } (\langle a, f\rangle \in \alpha \text{ and } \langle a, g\rangle \in \beta \text{ and } f(d) = g(d) = 1)\}$.

$N' = \mathrm{AddOp}[(G'_{m,\gamma})_{\gamma\in\{x,y,z\}}](N) = \langle (in_m[L_\tau]_\tau)_{\tau\in T}, (G'_{m,\gamma})_{\gamma\in\Gamma\cup\{x,y,z\}}\rangle$ is an algebra of the same signature ω' as B'. Finally, in_m is a (σ,ρ)-epimorphism from B to N and a (σ',ρ')-epimorphism from B' to N', where $\sigma' = \sigma$ and $\rho' = \rho \cup \{\langle x,x\rangle, \langle y,y\rangle, \langle z,z\rangle\}$.

3.2 AddSorts

An application of AddSorts to a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ has the form Addsorts$[S', \kappa](A)$, where S' is a set of sorts such that $S \cap S' = \emptyset$, and κ is a function from $S \cup S'$ to S such that $\kappa(s) = s$ for $s \in S$.

The resulting algebra $A' = \text{Addsorts}[S', \kappa](A) = \langle (A'_s)_{s\in(S\cup S')}, (F_\gamma)_{\gamma\in\Gamma'}\rangle$ has the set of sorts $S \cup S'$. For each sort $s \in S \cup S'$, the carrier A'_s of $\text{Addsorts}[S', \kappa](A)$ is identical to the carrier $A_{\kappa(s)}$ of A. So, AddSorts introduces new sorts without introducing new elements: the new sorts inherit the carriers of existing ones. The set Γ of operators of A is 'multiplied' accordingly: for all $\gamma \in \Gamma$ and $\langle\langle s_1, \ldots, s_n\rangle, s\rangle \in (S \cup S')^n \times (S \cup S')$ such that $\pi(\gamma) = \langle\langle\kappa(s_1), \ldots, \kappa(s_n)\rangle, \kappa(s)\rangle$, the algebra A' contains an operator $F_{\gamma\mapsto\langle\langle s_1,\ldots,s_n\rangle,s\rangle}$ of type $\langle\langle s_1, \ldots, s_n\rangle, s\rangle$, where $F_{\gamma\mapsto\langle\langle s_1,\ldots,s_n\rangle,s\rangle}(a_1, \ldots, a_n) = F_\gamma(a_1, \ldots, a_n)$ for all $\langle a_1, \ldots, a_n\rangle \in A_{\kappa(s_1)} \times \ldots \times A_{\kappa(s_n)}$. Thus each operator $F_{\gamma\mapsto\langle\langle s_1,\ldots,s_n\rangle,s\rangle}$ in A' is a replica of F_γ in A. Note that the set Γ' of operator indices of $\text{AddSorts}[S', \kappa](A)$ consists of compound symbols $\gamma \mapsto \langle\langle s_1, \ldots, s_n\rangle, s\rangle$, where γ is an operator index of A and $\langle\langle s_1, \ldots, s_n\rangle, s\rangle$ is the type of $F_{\gamma\mapsto\langle\langle s_1,\ldots,s_n\rangle,s\rangle}$. Hence $\text{AddSorts}[S', \kappa](A)$ has the type-assigning function $\{\langle\gamma \mapsto \vec{\gamma}, \vec{\gamma}\rangle \mid \gamma \mapsto \vec{\gamma} \in \Gamma'\}$.

The safeness of AddSorts can be proven in a straightforward manner. It entails that if $h \in \text{Epi}(A, B, \sigma, \rho)$,[30] where A and B are the π-algebras $\langle(A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ and $\langle(B_s)_{s\in S}, (G_\gamma)_{\gamma\in\Gamma}\rangle$, then there is a unique algebra B' such that $h \in \text{Epi}(A', B', \sigma', \rho')$[31] for $A' = \text{AddSorts}[S', \kappa](A)$: viz., $B' = \text{AddSorts}[S', \kappa](B)$.

AddSorts can be used for introducing new sorts which get their elements from one single old sort. For example, in the fragment of English, the expressions *man*, *walk* and *talks* are of different categories, viz., CN, V and IV, respectively. But MAN, WALK and TALK, the logical translations assigned to these expressions, are all of the same type (e, t).

In order to obtain similarity between the English algebra and the logical algebra, we will have to introduce two new sorts (V and IV, say) for the members of the carrier $L_{(e,t)}$ into the logical algebra. This can be achieved by deriving the algebra $B'' = \text{AddSorts}[\{\text{V,IV}\}, \mu](B')$ from the ω'-algebra $B' = \text{AddOp}[(F_\gamma)_{\gamma\in\{x,y,z\}}](B)$, where $\mu(\text{V}) = \mu(\text{IV}) = (e, t)$ and $\mu(\tau) = \tau$ for $\tau \in T$.

$\text{AddSorts}[\{\text{V,IV}\}, \mu](B')$ has the sorts $T \cup \{\text{V,IV}\}$ and the family of carriers $(L_\tau)_{\tau\in T} \cup \{L_{\text{V}}, L_{\text{IV}}\}$, where $L_{\text{V}} = L_{\text{IV}} = L_{(e,t)}$.

The set Γ' of operator indices of B'' is rather complex. It is listed below,

[30] As announced in footnote 29, we assume that σ and ρ (as well as the bijections σ' and ρ' from A' to B') are identity functions, so that A and B (and A' and B') have the same sorts, the same operator indices, and the same type-assigning function.

[31] The addition of sorts to an algebra does not affect the members of the carriers. So, again, the restriction h' of h to A' is the same function as h.

with some overlap between the specified subsets (the symbol ET denotes the set $\{\mathrm{V}, \mathrm{IV}, (e,t)\}$; the overlap is due to the fact that $ET \cap T = \{(e,t)\}$):

$$
\begin{aligned}
\Gamma' \;=\; & \{1 \mapsto \langle\langle t\rangle, t\rangle\} \cup \\
& \{2 \mapsto \langle\langle t,t\rangle, t\rangle\} \cup \\
& \{3{:}\tau{:}i \mapsto \langle\langle t\rangle, t\rangle \mid \tau \in T,\, i \in \mathbb{N}\} \cup \\
& \{4{:}\sigma{:}\tau \mapsto \langle\langle(\sigma,\tau),\sigma\rangle,\tau\rangle \mid \sigma \in T,\, \tau \in T\} \cup \\
& \{4{:}e{:}t \mapsto \langle\langle(\sigma), e\rangle, t\rangle \mid \sigma \in ET\} \cup \\
& \{4{:}(e,t){:}\tau \mapsto \langle\langle((e,t),\tau),\sigma\rangle,\tau\rangle \mid \sigma \in ET,\, \tau \in T\} \cup \\
& \{4{:}\sigma{:}(e,t) \mapsto \langle\langle(\sigma,(e,t)),\sigma\rangle,\tau\rangle \mid \sigma \in T,\, \tau \in ET\} \cup \\
& \{4{:}(e,t){:}(e,t) \mapsto \langle\langle((e,t),(e,t)),\sigma\rangle,\tau\rangle \mid \sigma \in ET,\, \tau \in ET\} \\
& \{5{:}\sigma{:}\tau{:}i \mapsto \langle\langle\tau\rangle,(\sigma,\tau)\rangle \mid \sigma \in T,\, \tau \in T\} \cup \\
& \{5{:}e{:}t{:}i \mapsto \langle\langle t\rangle,\tau\rangle \mid \tau \in ET\} \cup \\
& \{5{:}\sigma{:}(e,t){:}i \mapsto \langle\langle\tau\rangle,(\sigma,(e,t))\rangle \mid \sigma \in T,\, \tau \in ET\} \cup \\
& \{6{:}\tau \mapsto \langle\langle\tau,\tau\rangle,t\rangle \mid \tau \in T\} \cup \\
& \{6{:}(e,t) \mapsto \langle\langle\sigma,\tau\rangle,t\rangle \mid \sigma \in ET,\, \tau \in ET\} \cup \\
& \{x \mapsto \langle\langle\sigma\rangle,\tau\rangle \mid \sigma \in ET,\, \tau \in ET\} \cup \\
& \{y \mapsto \langle\langle\sigma,((e,t),(e,t))\rangle,\tau\rangle \mid \sigma \in ET,\, \tau \in ET\} \cup \\
& \{z \mapsto \langle\langle\rho,\sigma\rangle,\tau\rangle \mid \rho \in ET,\, \sigma \in ET,\, \tau \in ET\}
\end{aligned}
$$

For all $\gamma \mapsto \langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle \in \Gamma'$, the operator $F_{\gamma\mapsto\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle}$ has type $\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle$, and $F_{\gamma\mapsto\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle}(a_1,\ldots,a_n) = F_\gamma(a_1,\ldots,a_n)$ for every $\gamma \in \Gamma \cup \{x,y,z\}$ and $\langle a_1,\ldots,a_n\rangle \in L_{\mu(\tau_1)} \times \ldots \times L_{\mu(\tau_n)}$.

When it is applied to the algebras $B' = \mathrm{AddOp}[(F_\gamma)_{\gamma\in\{x,y,z\}}](B)$, $N' = \mathrm{AddOp}[(G'_{m,\gamma})_{\gamma\in\{x,y,z\}}](N)$ and the epimorphism in_m from B' to N', the safeness of AddSorts guarantees that there is a unique algebra N'' such that $in_m \in \mathrm{Epi}(\mathrm{AddSorts}[\{\mathrm{V},\mathrm{IV}\},\mu](B'), N'')$. Algebra N'' is the algebra $\mathrm{AddSorts}[\{\mathrm{V},\mathrm{IV}\},\mu](N')$, and it has the sorts $T \cup \{\mathrm{V},\mathrm{IV}\}$, the carriers $(in_m[L_{\mu(\tau)}]_\tau)_{\tau\in(T\cup\{\mathrm{V},\mathrm{IV}\})}$ and the above set Γ' of operator indices, where $G'_{m,\gamma\mapsto\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle}$ is of type $\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle$ and $G'_{m,\gamma\mapsto\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle}(a_1,\ldots,a_n) = G'_{m,\gamma}(a_1,\ldots,a_n)$ for all $\langle a_1,\ldots,a_n\rangle \in in_m[L_{\mu(\tau_1)}] \times \ldots \times in_m[L_{\mu(\tau_n)}]$.

While the function in_m is a (σ',ρ')-epimorphism from B' to N', it is a (σ'',ρ'')-epimorphism from B'' to N'', where $\sigma'' = \{\langle\tau,\tau\rangle \mid \tau \in (T\cup\{\mathrm{V},\mathrm{IV}\})\}$ and $\rho'' = \{\langle\gamma,\gamma\rangle \mid \gamma \in \Gamma'\}$.

3.3 DelOp

An application of DelOp to a π-algebra $A = \langle(A_s)_{s\in S},(F_\gamma)_{\gamma\in\Gamma}\rangle$ has the form $\mathrm{DelOp}[(F_\gamma)_{\gamma\in\Gamma'}](A)$, where Γ' is a set of operator indices such that $\Gamma' \subseteq \Gamma$.

The resulting algebra $\mathrm{DelOp}[(F_\gamma)_{\gamma\in\Gamma'}](A) = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in(\Gamma-\Gamma')}\rangle$ is of signature $\pi' = \pi - \{\langle\gamma,\pi(\gamma)\rangle \mid \gamma\in\Gamma'\}$. So, DelOp simply removes the operators in $(F_\gamma)_{\gamma\in\Gamma'}$ from A.

The safeness of the deriver DelOp is straightforward, and it entails that if $h\in \mathrm{Epi}(A,B,\sigma,\rho)$, where A and B are the π-algebras $\langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ and $\langle (B_s)_{s\in S}, (G_\gamma)_{\gamma\in\Gamma}\rangle$, then there is a unique algebra B' with the property that $h\in \mathrm{Epi}(A',B',\sigma',\rho')$[32] for $A' = \mathrm{DelOp}[(F_\gamma)_{\gamma\in\Gamma'}](A)$: viz., $B' = \mathrm{DelOp}[(G_\gamma)_{\gamma\in\Gamma'}](B)$.

DelOp is needed because the derived logical algebra $D(B)$ we are aiming at should have only operators which correspond to operators in $T_{A,K}$, the term algebra of the fragment of English. Observe that the algebra B'' we derived using AddSorts still has infinitely many operators, whereas only the following six are needed as counterparts of the respective operators H_1^T, H_2^T, H_3^T, H_4^T, H_5^T and H_6^T in $T_{A,K}$:

- $F_{x\mapsto\langle\langle \mathrm{V}\rangle,\mathrm{IV}\rangle}$,
- $F_{4:(e,t):t\mapsto\langle\langle\langle(e,t),t\rangle,\mathrm{IV}\rangle,t\rangle}$,
- $F_{4:(e,t):((e,t),t)\mapsto\langle\langle\langle(e,t),((e,t),t)\rangle,(e,t)\rangle,((e,t),t)\rangle}$,
- $F_{y\mapsto\langle\langle \mathrm{IV},((e,t),(e,t))\rangle,\mathrm{IV}\rangle}$,
- $F_{2\mapsto\langle\langle t,t\rangle,t\rangle}$, and
- $F_{z\mapsto\langle\langle \mathrm{IV},\mathrm{IV}\rangle,\mathrm{IV}\rangle}$.

Abbreviate the indices of these operators as a, b, c, d, e and f, respectively, and let Γ'' denote the set $\{a,b,c,d,e,f\}$.[33] In order to obtain similarity between the English term algebra and the logical algebra, it is necessary to remove all operators from $B'' = \mathrm{AddSorts}[\{\mathrm{V},\mathrm{IV}\},\mu](B')$, except the ones listed above. This can be achieved by deriving the algebra $B''' = \mathrm{DelOp}[(F_\gamma)_{\gamma\in(\Gamma'-\Gamma'')}](B'') = \langle (L_\tau)_{\tau\in T}\cup\{L_{\mathrm{V}},L_{\mathrm{IV}}\},(F_\gamma)_{\gamma\in\Gamma''}\rangle$ from B''. The

[32] The deletion of operators from an algebra does not affect the carriers. So, yet again, the restriction h' of h to A' is the same function as h.

[33] Note that Γ'' is a subset of Γ', the 'rather complex' set of operator indices listed in section 3.2 above.

type-assigning function ω''' of B''' is given by:

$$\omega'''(a) = \langle\langle \mathrm{V}\rangle, \mathrm{IV}\rangle,$$
$$\omega'''(b) = \langle\langle((e,t),t), \mathrm{IV}\rangle, t\rangle,$$
$$\omega'''(c) = \langle\langle((e,t),((e,t),t)),(e,t)\rangle,((e,t),t)\rangle,$$
$$\omega'''(d) = \langle\langle \mathrm{IV},((e,t),(e,t))\rangle, \mathrm{IV}\rangle,$$
$$\omega'''(e) = \langle\langle t,t\rangle, t\rangle, \text{ and}$$
$$\omega'''(f) = \langle\langle \mathrm{IV}, \mathrm{IV}\rangle, \mathrm{IV}\rangle.$$

When it is applied to the algebras $B'' = \mathrm{AddSorts}[\{\mathrm{V},\mathrm{IV}\},\mu](B')$, $N'' = \mathrm{AddSorts}[\{\mathrm{V},\mathrm{IV}\},\mu](N')$ and the epimorphism in_m from B'' to N'', the safeness of DelOp guarantees that there is a unique algebra N''' such that $in_m \in \mathrm{Epi}(\mathrm{DelOp}[(F_\gamma)_{\gamma\in(\Gamma'-\Gamma'')}](B''), N'')$. This algebra N''' is the algebra $\mathrm{DelOp}[(G'_{m,\gamma})_{\gamma\in(\Gamma'-\Gamma'')}](N'') = \langle(in_m[L_{\mu(\tau)}]_\tau)_{\tau\in(T\cup\{\mathrm{V},\mathrm{IV}\})}, (G'_{m,\gamma})_{\gamma\in\Gamma''}\rangle$.

The function in_m is a (σ'',ρ'')-epimorphism from B'' to N'', but it is a (σ''',ρ''')-epimorphism from B''' to N''', where $\rho''' = \{\langle\gamma,\gamma\rangle \mid \gamma\in\Gamma''\}$ and $\sigma''' = \sigma''$.

3.4 DelSorts

The fourth basic deriver is DelSorts.[34] An application of this deriver to an algebra $A = \langle(A_s)_{s\in S},(F_\gamma)_{\gamma\in\Gamma}\rangle$ of signature π has the form $\mathrm{DelSorts}[S'](A)$,

[34]DelSorts replaces the deriver SubAlg proposed by Janssen. An application of SubAlg to a π-algebra $A = \langle(A_s)_{s\in S},(F_\gamma)_{\gamma\in\Gamma}\rangle$ has the form $\mathrm{SubAlg}[H](A)$, were $H = (H_s)_{s\in S}$ is a sorted collection of sets H_s such that $H_s \subseteq A_s$ for all $s \in S$. Its effect is that an algebra is formed which has a subset of H as a generating set: $\mathrm{SubAlg}_H(A) = \langle[(H_\sigma)_{\sigma\in\Sigma}],(F'_\gamma)_{\gamma\in\Gamma'}\rangle$, where:

- $\Sigma = \{s \in S \mid H_s \neq \emptyset\}$,
- $\Gamma = \{\gamma\in\Gamma \mid \text{for some } n\in\mathbb{N}^+ : \pi(\gamma)\in\Sigma^n\times\Sigma\}$, and
- F'_γ is the restriction of F_γ to $[(H_\sigma)_{\sigma\in\Sigma}]$.

'This deriver is used in the following kind of situation. The logical algebra L [see (39) above] has for each sort infinitely many generators. The grammar A might not have this property. For instance, the sentences of a natural language are all built up from smaller components, and hence there are no generators of the sort 'sentence'. SubAlg is then used to reduce the carriers of L to those elements which will be images of elements in A.' (Janssen (1986), p. 83).

Note, however, that the deriver SubAlg has a destructive side-effect if it is employed in such a situation: SubAlg *deletes* a sort s of L if H_s is empty, with the unwelcome result that the sort in A will be deprived of a corresponding sort in L! Moreover, it is not clear why one would need a deriver which does something like this: whether the logical algebra L contains generators which are not images of elements in the grammar A is irrelevant for the possibility of there being a homomorphism from (the term algebra of) A to L. On the other hand, deletion of sorts is exactly what we need as a last step to obtain a derived

where S' is a subset of S such that if $s \in S'$ and if for some γ it holds that $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then $s \notin \{s_1, \ldots, s_{n+1}\}$.

The resulting algebra $\mathrm{DelSorts}[S'](A) = \langle (A_s)_{s\in(S-S')}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ is of the same signature as A. So, DelSorts removes sorts and their carriers, provided that these sorts do not figure in the types of the operators of A.

As a consequence of the (straightforward) safeness of this deriver, we have that if $h \in \mathrm{Epi}(A, B, \sigma, \rho)$ for π-algebras $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ and $B = \langle (B_s)_{s\in S}, (G_\gamma)_{\gamma\in\Gamma}\rangle$, then there is a unique algebra B' with the property that $h' \in \mathrm{Epi}(A', B', \sigma', \rho')$, where h' is the restriction[35] of h to $A' = \mathrm{DelSorts}[T'](A)$: viz., $B' = \mathrm{DelSorts}[T'](B)$.

For instance, consider the ω'''-algebra B''' that was derived using DelOp, viz., $B''' = \mathrm{DelOp}[(F_\gamma)_{\gamma\in(\Gamma'-\Gamma'')}](B'') = \langle (L_\tau)_{\tau\in T} \cup \{L_{\mathrm{V}}, L_{\mathrm{IV}}\}, (F_\gamma)_{\gamma\in\Gamma''}\rangle$. The type-assigning function ω''' of B''' was defined in the previous section. Using DelSorts, we can delete all sorts that do *not* occur in some $\omega'''(\gamma)$. If T' denotes the set consisting of these sorts, then $((T \cup \{\mathrm{V,IV}\}) - T')$ is the following set:

$$\{t,\ (e,t),\ \mathrm{V},\ \mathrm{IV},\ ((e,t),t),\ ((e,t),((e,t),t)),\ ((e,t),(e,t))\}$$

And $B'''' = \mathrm{DelSorts}[T'](B''')$ is the following algebra:

$$\langle (L_\tau)_{\tau\in((T\cup\{\mathrm{V,IV}\})-T')}, (F_\gamma)_{\gamma\in\Gamma''}\rangle$$

When it is applied to the algebras $B''' = \mathrm{DelOp}[(F_\gamma)_{\gamma\in(\Gamma'-\Gamma'')}]$, $N''' = \mathrm{DelOp}[(G'_{m,\gamma})_{\gamma\in(\Gamma'-\Gamma'')}](N'')$ and the epimorphism in_m from B''' to N''', the safeness of DelSorts guarantees that there is a unique algebra N'''' with the property that for the restriction in'_m of in_m to $B'''' = \mathrm{DelSorts}[T'](B''')$ it holds that $in'_m \in \mathrm{Epi}(B'''', N'''')$, viz.: $N'''' = \mathrm{DelSorts}[T'](N''') =$

$$\langle (in_m[L_{\mu(\tau)}]_\tau)_{\tau\in((T\cup\{\mathrm{V,IV}\})-T')}, (G'_{m,\gamma})_{\gamma\in\Gamma''}\rangle$$

While in_m is a (σ''', ρ''')-epimorphism from B''' to N''', the function in'_m is a (σ'''', ρ'''')-epimorphism from B'''' to N'''', where $\rho'''' = \rho'''$ and $\sigma'''' = \{\langle\tau,\tau\rangle \mid \tau \in ((T \cup \{\mathrm{V,IV}\}) - T')\}$.

---

logical algebra which is similar to A. Instead of SubAlg, we therefore propose the more simple deriver DelSorts.

[35] Since DelSorts deletes carriers, the restriction h' of h to A' is not necessarily the same function as h.

Besides, by deriving B'''' we have at last obtained an algebra which is (σ,ρ)-similar to the English term algebra $T_{A,K}$: (*i*) the sorts of $T_{A,K}$ and B'''' are connected by the bijection $\sigma : C \rightarrow ((T \cup \{\mathrm{V},\mathrm{IV}\}) - T')$, where

$$\begin{array}{lll} \sigma(\mathrm{S}) = t, & \sigma(\mathrm{Adv}) = ((e,t),(e,t)), & \sigma(\mathrm{Det}) = ((e,t),((e,t),t)), \\ \sigma(\mathrm{CN}) = (e,t), & \sigma(\mathrm{T}) = ((e,t),t), & \sigma(\mathrm{V}) = \mathrm{V}, \quad \sigma(\mathrm{IV}) = \mathrm{IV}; \end{array}$$

(*ii*) the operator indices of $T_{A,K}$ and B'''' are connnected by the bijection $\rho : \Delta \rightarrow \{a,b,c,d,e,f\}$, where

$$\begin{array}{l} \rho(1) = a = x \mapsto \langle\langle \mathrm{V}\rangle, \mathrm{IV}\rangle, \\ \rho(2) = b = 4{:}(e,t){:}t \mapsto \langle\langle((e,t),t),\mathrm{IV}\rangle,t\rangle, \\ \rho(3) = c = 4{:}(e,t){:}((e,t),t) \mapsto \langle\langle((e,t),((e,t),t)),(e,t)\rangle,((e,t),t)\rangle, \\ \rho(4) = d = y \mapsto \langle\langle \mathrm{IV},((e,t),(e,t))\rangle,\mathrm{IV}\rangle, \\ \rho(5) = e = 2 \mapsto \langle\langle t,t\rangle,t\rangle, \\ \rho(6) = f = z \mapsto \langle\langle \mathrm{IV},\mathrm{IV}\rangle,\mathrm{IV}\rangle; \text{ and} \end{array}$$

(*iii*) for all $\delta \in \Delta$: if H_δ^T is assigned type $\langle\langle c_1,\ldots,c_n\rangle,c\rangle$ in $T_{A,K}$, then $F_{\rho(\delta)}$ is assigned type $\langle\langle\sigma(c_1),\ldots,\sigma(c_n)\rangle,\sigma(c)\rangle$ in B''''.

And finally, the translation *tr* defined in section 2 is a (σ,ρ)-homomorphism from $T_{A,K}$ to B'''', since $tr[T_{A,K,c}] \subseteq L_{\sigma(c)}$ for all $c \in C$; and for all $\delta \in \Delta$ we have that if $\pi(\delta) = \langle\langle c_1,\ldots,c_n\rangle,c\rangle$ and $t_1 \in T_{A,K,c_1}$, $\ldots, t_n \in T_{A,K,c_n}$, then $tr(H_\delta^T(t_1,\ldots,t_n)) = F_{\rho(\delta)}(tr(t_1),\ldots,tr(t_n))$.

4 Interpretability

Thus, using the basic derivers AddOp, AddSorts, DelOp and DelSorts, it turns out to be possible to derive the required similar algebra

$$B'''' = \mathrm{DelSorts}(\mathrm{DelOp}(\mathrm{AddSorts}(\mathrm{AddOp}(B))))$$

in four steps from the original logical algebra B. But is it also necessary to follow this laborious process? Janssen emphasizes repeatedly that the possibility of a homomorphism presupposes similarity: 'A mapping is called a homomorphism if it respects the structures of the algebras involved. This is only possible if the two algebras have a similar structure.' (Janssen (1986), Part 1, pp. 21–22; cf. also pp. 67–70).

Nevertheless, we observed above that if, instead of similarity, interpretability is assumed, we are done in one step: when algebra B is enriched

with operators $F_x : L_{(e,t)} \rightarrow L_{(e,t)}$, $F_y : L_{(e,t)} \times L_{((e,t),(e,t))} \rightarrow L_{(e,t)}$ and $F_z : L_{(e,t)} \times L_{(e,t)} \rightarrow L_{(e,t)}$, where

$$F_x(\alpha) = \alpha,$$
$$F_y(\alpha, \beta) = F_{4:(e,t):(e,t)}(\beta, \alpha), \text{ and}$$
$$F_z(\alpha, \beta) = F_{5:e:t:0}(F_2(F_{4:e:t}(\alpha, v_{0,e}), F_{4:e:t}(\beta, v_{0,e}))),$$

then $T_{A,K}$ is (σ, ρ)-interpretable[36] in the resulting algebra B', and the function *tr* defined in (18) is a (σ, ρ)-homomorphism from $T_{A,K}$ to B'.

As regards the aspect of deriving a new syntactic algebra from the syntactic algebra of the logical language, Janssen notes:

> In one respect this attempt [to formalize the compositionality principle] probably has not been successful: the description of how to obtain new algebras out of old ones. There is no general theory which I could use here, and I had to apply *ad hoc* methods. (Janssen (1986), Part 1, p. 42; see also p. 83)

It seems, however, justified to conclude that the very notion of a 'safe deriver' is *ad hoc*, since it is an artefact created by the requirement of similarity – a requirement which, as we argued above, is itself undermotivated in view of the conditions imposed by the compositionality principle. Accordingly, we will show below that the addition of operators to the logical algebra is not, as Janssen puts it, the 'most important' deriver, but the only 'deriver' that has to be taken into account at all.

In order to demonstrate this, we will first go into the role of a formal logical language as an auxiliary translation language.

The basic idea is simply that a term in the term algebra of the generated syntactic algebra is translated into an expression of the logical language, and that it is indirectly assigned the semantic interpretation of the logical expression which serves as its translation. (And an expression in the syntactic algebra is assigned the interpretations of the translations of those terms of

[36] I.e., for $\sigma : C \rightarrow T$ and $\rho : \Delta \rightarrow \Gamma \cup \{x, y, z\}$ such that

- $\sigma(\text{S}) = t$, $\sigma(\text{CN}) = \sigma(\text{V}) = \sigma(\text{IV}) = (e,t)$, $\sigma(\text{T}) = ((e,t),t)$, $\sigma(\text{Det}) = ((e,t),((e,t),t))$ and $\sigma(\text{Adv}) = ((e,t),(e,t))$; and
- $\rho(1) = x$, $\rho(2) = 4{:}(e,t){:}t$, $\rho(3) = 4{:}(e,t){:}((e,t),t)$, $\rho(4) = y$, $\rho(5) = 2$ and $\rho(6) = z$.

which it is the evaluation.) Thus, 'the principal use of translations is the semantical one of inducing interpretations' (Montague (1970b), p. 232).

In general, formal logical languages and their semantic interpretations are defined by specifying (*i*) a generated algebra of well-formed logical expressions; and (*ii*) an interpretation homomorphism from this generated algebra to a semantic algebra. This homomorphism is not specified by stating its values for all arguments (there are generally infinitely many well-formed expressions), but by (*a*) providing a mapping $\mathcal{I}$ which assigns a member of sort $\sigma(\tau)$ in the semantic algebra to each generator of sort τ in the logical algebra; and (*b*) associating each logical operator F_γ of type $\langle\langle\tau_1,\ldots,\tau_n\rangle,\tau\rangle$ with a semantic operator $G_{\rho(\gamma)}$ of type $\langle\langle\sigma(\tau_1),\ldots,\sigma(\tau_n)\rangle,\sigma(\tau)\rangle$, whereby $\mathcal{I}(F_\gamma(\alpha_1,\ldots,\alpha_n))$ is defined as $G_{\rho(\gamma)}(\mathcal{I}(\alpha_1),\ldots,\mathcal{I}(\alpha_n))$.[37]

Note that this procedure is only guaranteed to result in a *homomorphism* $\mathcal{I}$ if the logical language is syntactically unambiguous.[38] Therefore, the generated algebra of a logical language is as a rule a *free* algebra, in conformity with Gamut's adage that 'logical languages wear their meanings on their sleeves'.[39]

Furthermore, formal logical languages L usually have a *model-theoretic* interpretation, which means that their interpretation homomorphism $\mathcal{I}$ is defined pointwise: on the basis of a class[40] $\mathcal{M}$ of models for L, the interpre-

[37]The correspondence between the sorts and operator indices of the logical and the semantic algebra is usually established by bijections (identity functions) σ and ρ. We will henceforth assume that σ and ρ are identity functions.

[38]Note that $\mathcal{I}$ is designed so as to respect sorts and operators. If the logical language is syntactically unambiguous, then all generators belong to exactly one sort and are different from all non-generators, so that every generator is assigned exactly one value by $\mathcal{I}$. Moreover, the operators are injections with disjoint ranges, which means that also the non-generators receive a unique value, and, consequently, that $\mathcal{I}$ is a homomorphism.

[39]L.T.F. Gamut, p.c.

[40]Here the word 'class' is used deliberately rather than 'set', since the collection of models for a logical language is generally not a set in the sense of axiomatic set theory. In Section 1 above, for example, the models for the typed logical language L were defined in (6) through (8). Observe that each non-empty set E gives rise to a distinct domain $D_{E,e}$, so that there are at least as many frames – and, consequently, models – as there are (non-empty) sets. This means that the collection $\mathcal{M}$ of models for L is too large to be countenanced as a set: it is a proper class. Moreover, if $\mathcal{M}$ is a proper class, then the interpretations $\mathcal{I}(\alpha)$ which will be defined below must be proper classes as well: these collections contain for all $m \in \mathcal{M}$ exactly one ordered pair $\langle m, a\rangle$ and are, hence, just as large as $\mathcal{M}$. Finally, proper classes do not correspond to set-theoretical objects, so they cannot be constituents of sets and ordered pairs (which are a special kind of sets). Therefore, also the objects $\mathcal{I}_\tau$, $\mathcal{G}_\gamma$ and $\mathcal{S}$, which will be defined in terms of $\mathcal{I}(\alpha)$ below, do not necessarily correspond to sets. (The use of calligraphic letters for these notions is

tation $in_m(\alpha)$ of an expression $\alpha \in L$ in $m \in \mathcal{M}$ is given by

(*a*) a specification of $in_m(\alpha)$ for generators α of L; and

(*b*) an assignment of a semantic operator $G_{m,\gamma}$ to each logical operator F_γ of L, where $in_m(F_\gamma(\alpha_1, \ldots, \alpha_n))$ for expressions $F_\gamma(\alpha_1, \ldots, \alpha_n)$ is defined as $G_{m,\gamma}(in_m(\alpha_1), \ldots, in_m(\alpha_n))$.

Of course, the point of this model-theoretic set-up is that an expression can have different interpretations in different models: there is not in general a single object that serves as the interpretation of an expression α in all models m. In keeping with this, the standard definition of logical equivalence has it that two expressions α and β are logically equivalent – i.e., 'have the same logical meaning' (Gamut (1991), Vol. I, p. 49) – if and only if for all $m \in \mathcal{M}$ it holds that $in_m(\alpha) = in_m(\beta)$.

Consequently, in order to be able to talk about 'the' interpretation $\mathcal{I}(\alpha)$ of an expression α, one has to incorporate the models into the concept of interpretation: $\mathcal{I}(\alpha)$ is that function from models to interpretations in models such that $\mathcal{I}(\alpha)(m) = in_m(\alpha)$ for all $m \in \mathcal{M}$.[41]

If $B = \langle (L_\tau)_{\tau \in T}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ is a model-theoretically interpreted logical algebra of signature π, then the interpretation function $\mathcal{I}$ can be construed as an epimorphism from B to a semantic algebra $\mathcal{S}$ of the same signature, viz., $\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau \in T}, (\mathcal{G}_\gamma)_{\gamma \in \Gamma} \rangle$, where

- $\mathcal{I}_\tau = \{\mathcal{I}(\alpha) \mid \alpha \in L_\tau\}$; and
- if $\pi(\gamma) = \langle \langle \tau_1, \ldots \tau_n \rangle, \tau \rangle$, then $\mathcal{G}_\gamma : \mathcal{I}_{\tau_1} \times \ldots \times \mathcal{I}_{\tau_n} \to \mathcal{I}_\tau$, where $\mathcal{G}_\gamma(\mathcal{I}(\alpha_1), \ldots, \mathcal{I}(\alpha_n)) = \mathcal{I}(\alpha)$ iff for all $m \in \mathcal{M}$: $G_{m,\gamma}(in_m(\alpha_1), \ldots, in_m(\alpha_n)) = in_m(\alpha)$.

We have that $\mathcal{I}_\tau = \{\mathcal{I}(\alpha) \mid \alpha \in L_\tau\}$, so $\mathcal{I}$ respects the sorts in a surjective way: $\mathcal{I}[L_\tau] = \mathcal{I}_\tau$. Moreover, because it holds for all $m \in \mathcal{M}$

meant to visualize the set-theoretical proviso of the present footnote.)

[41] For example, when applied to the clauses (6) through (10), this definition yields that the interpretation $\mathcal{I}(\alpha)$ of an expression α of type τ in the language L of Section 1 is a collection of ordered pairs $\langle m, in_m(\alpha) \rangle$, where $m \in \mathcal{M}$ is a model $\langle (D_{E,\tau})_{\tau \in T}, I \rangle$ and $in_m(\alpha) \in D^A_{E,\tau}$ is a function from variable assignments to members of the set $D_{E,\tau}$.

Note that the observation made in footnote 40 entails that $\mathcal{I}(\alpha)$ cannot be viewed as a function in the strict set-theoretic sense: it is not a set, but a *proper class* of ordered pairs such that if $\langle m, a \rangle \in \mathcal{I}(\alpha)$ and $\langle m, b \rangle \in \mathcal{I}(\alpha)$, then $a = b$.

that $in_m(F_\gamma(\alpha_1,\ldots,\alpha_n)) = G_{m,\gamma}(in_m(\alpha_1),\ldots,in_m(\alpha_n))$, we also have that $\mathcal{I}(F_\gamma(\alpha_1,\ldots,\alpha_n)) = G_\gamma(\mathcal{I}(\alpha_1),\ldots,\mathcal{I}(\alpha_n))$, so $\mathcal{I}$ respects the operators.

Recall that the basic idea of using a formal logical language as an auxiliary translation language is that a term in the term algebra of the generated syntactic algebra is indirectly assigned the interpretation of the logical expression which serves as the translation of the term. Thus, each syntactic term t is associated with a unique translation $tr(t)$, and this translation induces the interpretation $\mathcal{I}(tr(t))$.

So, assume that each member t of a carrier of the term algebra $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle$ of a syntactic π-algebra $A = \langle (E_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ with respect to the generating set $(K_c)_{c\in C}$ is to be assigned a member $tr(t)$ of a carrier of a logical ω-algebra $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ which is interpreted in a semantic ω-algebra $\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau\in T}, (\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$ via a (σ',ρ')-epimorphism $\mathcal{I}$, where the sorts and operator indices of B and $\mathcal{S}$ are connected via identity functions σ' and ρ' (cf. footnote 37).

Note that for such an indirect interpretation assignment to be compositional, the composition $tr\circ\mathcal{I}$ of the translation and interpretation function has to be a homomorphism. This entails that $tr\circ\mathcal{I}$ has to respect the sorts, so there must be a function σ from the sorts C of $T_{A,K}$ to the sorts T of $\mathcal{S}$ such that $tr\circ\mathcal{I}[T_{A,K,c}] \subseteq \mathcal{I}_{\sigma(c)}$ for all $c \in C$. However, since the correspondence between the sorts of B and $\mathcal{S}$ is established by an injection σ', this means that the translation tr by itself must also respect the sorts, i.e.: $tr[T_{A,K,c}] \subseteq L_{\sigma(c)}$ for all $c \in C$.

The assignment of translations $tr(t)$ to terms t in the term algebra of a generated syntactic algebra proceeds in a way analogous to the assignment of interpretations to expressions in a logical language: because there are generally infinitely many terms to be translated, the function tr is not specified by stating its values for all terms, but by (*a*) providing a mapping tr which assigns a member of $L_{\sigma(c)}$ in the logical algebra to each term $\lfloor k_c\rfloor$ which corresponds to a generator k of sort c in the syntactic algebra; and (*b*) associating each operator H^T_δ of type $\langle\langle c_1,\ldots,c_n\rangle, c\rangle$ in the term algebra with some function $F_\delta : L_{\sigma(c_1)} \times \ldots \times L_{\sigma(c_n)} \to L_{\sigma(c)}$, and defining $tr(\lfloor \underline{H}_\delta\, t_1 \ldots t_n\rfloor)$ as $F_\delta(tr(t_1),\ldots,tr(t_n))$.

Since the term algebra of a generated syntactic algebra is a free algebra, this procedure is guaranteed to result in the assignment of a unique translation $tr(t)$ to each term t.

Besides, the π-algebra $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle$ is (σ,ρ)-interpretable in the π'-algebra $B' = \langle (L_\tau)_{\tau\in T}, (F_\delta)_{\delta\in\Delta}\rangle$ for ρ and π' such that $\rho(\delta) =$

δ and $\pi'(\delta) = \langle\langle\sigma(c_1),\ldots\sigma(c_n)\rangle,\sigma(c)\rangle$ iff $\pi(\delta) = \langle\langle c_1,\ldots c_n\rangle, c\rangle$, and the translation function *tr* is a (σ,ρ)-homomorphism from $T_{A,K}$ to B'.

It is worth mentioning that the logical algebra is usually exploited 'at a higher level' in the process of translation. Thus, terms corresponding to generators of the syntactic algebra need not be translated into generators of the logical algebra. In Section 1, for example (see (13)), the generators of the syntactic sorts CN, V and Adv are assigned generators of the corresponding logical sorts (e,t), (e,t) and $((e,t),(e,t))$, but the generators of the syntactic sorts T and Det are translated into compound expressions of the corresponding logical sorts $((e,t),t)$ and $((e,t),((e,t),t))$. And, more importantly, the functions F_δ associated with the operators H^T_δ of the term algebra do not necessarily coincide with operators F_γ that are actually present in the logical algebra. In (18), for instance, the operators H^T_2, H^T_3 and H^T_5 in the term algebra turn out to correspond to operators in the logical algebra, viz., $F_{4:(e,t):t}$, $F_{4:(e,t):((e,t),t)}$ and F_2, respectively, but this does not hold for the operators H^T_1, H^T_4 and H^T_6.

Let $\mathcal{I}$ be the epimorphism from the logical algebra $B = \langle(L_\tau)_{\tau\in T},(F_\gamma)_{\gamma\in\Gamma}\rangle$ to the semantic algebra $\mathcal{S} = \langle(\mathcal{I}_\tau)_{\tau\in T},(\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$; let $L'_{\tau_1}\subseteq L_{\tau_1},\ldots,L'_{\tau_n}\subseteq L_{\tau_n}$, and $L'_\tau\subseteq L_\tau$; and let $F_\delta : L'_{\tau_1}\times\ldots\times L'_{\tau_n}\rightarrow L'_\tau$ be an arbitrary function (that is: δ is not necessarily a member of Γ). Then $F^{\mathcal{I}}_\delta$, *the relation $\mathcal{I}$-induced by* F_δ, is the following collection:

(47) $\{\langle\langle\mathcal{I}(\alpha_1),\ldots,\mathcal{I}(\alpha_n)\rangle,\mathcal{I}(\alpha)\rangle \mid \langle\langle\alpha_1,\ldots,\alpha_n\rangle,\alpha\rangle\in F_\delta\}$

Moreover, such a function F_δ is *$\mathcal{I}$-functional* iff $F^{\mathcal{I}}_\delta$ is a function, i.e., iff there are no $\langle\langle\varepsilon_1,\ldots,\varepsilon_n\rangle,\varepsilon\rangle\in F^{\mathcal{I}}_\delta$ and $\langle\langle\varepsilon'_1,\ldots,\varepsilon'_n\rangle,\varepsilon'\rangle\in F^{\mathcal{I}}_\delta$ such that $\langle\varepsilon_1,\ldots,\varepsilon_n\rangle = \langle\varepsilon'_1,\ldots,\varepsilon'_n\rangle$ while $\varepsilon\neq\varepsilon'$.

As for the compositionality of an indirect interpretation assignment in terms of a translation homomorphism *tr* from $T_{A,K} = \langle(T_{A,K,c})_{c\in C},(H^T_\delta)_{\delta\in\Delta}\rangle$ to $B' = \langle(L_\tau)_{\tau\in T},(F_\delta)_{\delta\in\Delta}\rangle$ and an interpretation epimorphism $\mathcal{I}$ from $B = \langle(L_\tau)_{\tau\in T},(F_\gamma)_{\gamma\in\Gamma}\rangle$ to $\mathcal{S} = \langle(\mathcal{I}_\tau)_{\tau\in T},(\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$, it can be observed that the composition $tr\circ\mathcal{I}$ of the translation and interpretation function is a homomorphism if and only if all functions F_δ associated with operators H^T_δ in the term algebra $T_{A,K}$ are $\mathcal{I}$-functional. This can be seen as follows.

On the one hand, suppose that all $F_\delta : L_{\sigma(c_1)}\times\ldots\times L_{\sigma(c_n)}\rightarrow L_{\sigma(c)}$ in algebra B' associated with operators H^T_δ of type $\langle\langle c_1,\ldots,c_n\rangle,c\rangle$ in the term algebra $T_{A,K}$ are $\mathcal{I}$-functional. Then all these functions $\mathcal{I}$-induce *functions*

$F_\delta^{\mathcal{I}}$, so that $\mathcal{S}' = \langle (\mathcal{I}_\tau)_{\tau\in T}, (F_\delta^{\mathcal{I}})_{\delta\in\Delta}\rangle$ is an algebra, and $\mathcal{I}$ is an epimorphism from B' to $\mathcal{S}'$, due to the way in which the $F_\delta^{\mathcal{I}}$ are defined (see (47)). Moreover, we can use the fact – established in Section 2 above – that the composition of two homomorphisms is again a homomorphism to conclude that $tr \circ \mathcal{I}$ is a homomorphism from $T_{A,K}$ to $\mathcal{S}'$.

On the other hand, if some $F_\delta : L_{\sigma(c_1)} \times \ldots \times L_{\sigma(c_n)} \to L_{\sigma(c)}$ in algebra B' associated with an operator H_δ^T of type $\langle\langle c_1, \ldots, c_n\rangle, c\rangle$ in the term algebra $T_{A,K}$ happens to $\mathcal{I}$-induce some non-functional *relation* $F_\delta^{\mathcal{I}}$, then $tr \circ \mathcal{I}$ can fail to be a homomorphism on account of its not being a function: in that event $F_\delta^{\mathcal{I}}$ contains $\langle\langle \mathcal{I}(\alpha_1), \ldots, \mathcal{I}(\alpha_n)\rangle, \varepsilon\rangle$ and $\langle\langle \mathcal{I}(\alpha_1), \ldots, \mathcal{I}(\alpha_n)\rangle, \varepsilon'\rangle$ such that $\varepsilon \neq \varepsilon'$. Hence if $\alpha_1 = tr(t_1), \ldots, \alpha_n = tr(t_n)$ for terms $t_1, \ldots, t_n$,[42] then there are two options for $\mathcal{I}(tr(H_\delta^T(t_1, \ldots, t_n))) = \mathcal{I}(F_\delta(tr(t_1), \ldots, tr(t_n))) = F_\delta^{\mathcal{I}}(\mathcal{I}(tr(t_1)), \ldots, \mathcal{I}(tr(t_n))) = F_\delta^{\mathcal{I}}(\mathcal{I}(\alpha_1), \ldots, \mathcal{I}(\alpha_n))$, viz., ε and ε'.

So, in order for the composition $tr \circ \mathcal{I}$ of a homomorphism tr from $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H_\delta^T)_{\delta\in\Delta}\rangle$ to $B' = \langle (L_\tau)_{\tau\in T}, (F_\delta)_{\delta\in\Delta}\rangle$ and an epimorphism $\mathcal{I}$ from $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ to $\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau\in T}, (\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$ to be a homomorphism from $T_{A,K}$ to $\mathcal{S}' = \langle (\mathcal{I}_\tau)_{\tau\in T}, (F_\delta^{\mathcal{I}})_{\delta\in\Delta}\rangle$, all functions F_δ associated with operators H_δ^T in $T_{A,K}$ must $\mathcal{I}$-induce a *function* $F_\delta^{\mathcal{I}}$.

Accordingly, the following question arises: given some epimorphism $\mathcal{I}$ from $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ to $\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau\in T}, (\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$, which operators $F_\delta : L_{\tau_1} \times \ldots \times L_{\tau_n} \to L_\tau$ are $\mathcal{I}$-functional?

Note that we already have a partial answer to this question. In Section 3.1 above, we showed that the addition to an algebra $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ of a set $(P_\delta^\#)_{\delta\in\Delta}$ of operators $P_\delta^\#$ determined by polynomial symbols P_δ over B[43] leads to an algebra $B' = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma} \cup (P_\delta^\#)_{\delta\in\Delta}\rangle$ with the property that for *every* algebra $C = \langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma}\rangle$ and epimorphism $\mathcal{I}$ from B to C, there is a (unique) algebra C' such that $\mathcal{I}$ is an epimorphism

[42]Note that, strictly speaking, it is not the $F_\delta : L_{\sigma(c_1)} \times \ldots \times L_{\sigma(c_n)} \to L_{\sigma(c)}$ themselves, but their restrictions $F_\delta|tr = F_\delta \cap ((tr[T_{A,K,s_1}] \times \ldots \times tr[T_{A,K,s_n}]) \times tr[T_{A,K,s}])$ which have to be $\mathcal{I}$-functional in order for $tr \circ \mathcal{I}$ to be a homomorphism. However, since every $\mathcal{I}$-functional function $F : tr[T_{A,K,s_1}] \times \ldots \times tr[T_{A,K,s_n}] \to tr[T_{A,K,s}]$ can be extended to a $\mathcal{I}$-functional function $F' : L_{\sigma(c_1)} \times \ldots \times L_{\sigma(c_n)} \to L_{\sigma(c)}$ (simpy avoid non-equivalent values for equivalent arguments outside $[T_{A,K,s_1}] \times \ldots \times tr[T_{A,K,s_n}]$), there is for every non-$\mathcal{I}$-functional $F_\delta : L_{\sigma(c_1)} \times \ldots \times L_{\sigma(c_n)} \to L_{\sigma(c)}$ with $\mathcal{I}$-functional $F_\delta|tr$ an $\mathcal{I}$-functional function $F'_\delta : L_{\sigma(c_1)} \times \ldots \times L_{\sigma(c_n)} \to L_{\sigma(c)}$ such that $F'_\delta|tr = F_\delta|tr$. The latter entails that an algebra $B' = \langle (L_\tau)_{\tau\in T}, (F_\delta)_{\delta\in\Delta}\rangle$ which mediates in an indirect interpretation homomorphism $tr \circ \mathcal{I}$ can always be replaced by an algebra $B'' = \langle (L_\tau)_{\tau\in T}, (F'_\delta)_{\delta\in\Delta}\rangle$ in which all operator are $\mathcal{I}$-functional.

[43]Recall that these *polynomial* operators $P_\delta^\#$ are (definable in terms of) projection functions, constant functions and the operators in $(F_\gamma)_{\gamma\in\Gamma}$.

from B' to C', viz., $C' = \langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma} \cup ((f(P_\delta))^\#)_{\delta\in\Delta}\rangle$, where f is a function from polynomial symbols over B to polynomial symbols over C such that $\mathcal{I}(P_\delta^\#(\alpha_1,\ldots,\alpha_n)) = (f(P_\delta))^\#(\mathcal{I}(\alpha_1),\ldots,\mathcal{I}(\alpha_n))$ for all $\alpha_1,\ldots,\alpha_n$. Of course, the latter means that $(f(P_\delta))^\# = (P_\delta^\#)^{\mathcal{I}}$, the relation $\mathcal{I}$-induced by $P_\delta^\#$. Consequently, polynomial operators over an algebra B are $\mathcal{I}$-functional for *all* algebras C and epimorphisms $\mathcal{I}$ from B to C.

However, given a *particular* epimorphism $\mathcal{I}$ from the logical algebra $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ to a *specific* semantic algebra $\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau\in T}, (\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$, there are always[44] non-polynomial $\mathcal{I}$-functional operators. For either there are no $\tau\in T$, $\alpha\in L_\tau$ and $\alpha'\in L_\tau$ such that $\mathcal{I}(\alpha) = \mathcal{I}(\alpha')$, or there are such $\tau\in T$, $\alpha\in L_\tau$ and $\alpha'\in L_\tau$. In the former – peculiar[45] – situation every operator F_δ is necessarily $\mathcal{I}$-functional, since then $\langle \mathcal{I}(\alpha_1),\ldots,\mathcal{I}(\alpha_n)\rangle = \langle \mathcal{I}(\alpha_1'),\ldots,\mathcal{I}(\alpha_n')\rangle$ entails that $\langle \alpha_1,\ldots,\alpha_n\rangle = \langle \alpha_1',\ldots,\alpha_n'\rangle$, so that $F_\delta^{\mathcal{I}}$ inherits its being a function from F_δ. In the latter situation, where for some $\tau\in T$, $\alpha\in L_\tau$ and $\alpha'\in L_\tau$ it holds that $\mathcal{I}(\alpha) = \mathcal{I}(\alpha')$, it can be noted that the operator $F_\delta : L_\tau \to L_\tau$ such that $F_\delta(\alpha) = \alpha'$, $F_\delta(\alpha') = \alpha$ and $F(\beta) = \beta$ for $\beta\in(L_\tau - \{\alpha,\alpha'\})$ is $\mathcal{I}$-functional but non-polynomial.[46]

---

[44] Well, almost always: some carrier L_τ of B will have to contain at least two expressions. For if all carriers of B contain at most one expression, then either:
(*a*) $L_{\tau_1}\times\ldots\times L_{\tau_n} = \{\langle\alpha_1,\ldots,\alpha_n\rangle\}$, $L_\tau = \emptyset$, and there is no $F_\delta : L_{\tau_1}\times\ldots\times L_{\tau_n}\to L_\tau$;
(*b*) $L_{\tau_1}\times\ldots\times L_{\tau_n} = \{\langle\alpha_1,\ldots,\alpha_n\rangle\}$, $L_\tau = \{\alpha\}$, and for $F_\delta : L_{\tau_1}\times\ldots\times L_{\tau_n}\to L_\tau$ we have that $F_\delta = \{\langle\langle\alpha_1,\ldots,\alpha_n\rangle,\alpha\rangle\}$, which is a constant function; or
(*c*) $L_{\tau_1}\times\ldots\times L_{\tau_n} = \emptyset$, and for $F_\delta : L_{\tau_1}\times\ldots\times L_{\tau_n}\to L_\tau$ we have that $F_\delta = \emptyset$, which is a projection function.

If, on the other hand, some carrier L_τ contains at least two expressions α and α', then any operator $F_\delta : L_\tau\to L_\tau$ such that $F_\delta(\alpha) = \alpha'$ and $F_\delta(\alpha') = \alpha$ is non-polynomial:
(*a*) F_δ is not a constant function, since $F_\delta(\alpha)\neq F_\delta(\alpha')$,;
(*b*) F_δ is not a projection function, since $F_\delta(\alpha)\neq\alpha$; and
(*c*) F_δ is not determined by a polynomial symbol $\lfloor \underset{\sim}{F}_\gamma \lfloor p_1\rfloor\ldots\lfloor p_n\rfloor\rfloor$ which contains the polynomial variable $\lfloor\xi_\tau^1\rfloor$, since for such F it holds that for all $\beta\in L_\tau$ the term $\lfloor\beta\rfloor^t$ is a proper subterm of $\lfloor F(\beta)\rfloor^t$, and $\lfloor\alpha\rfloor^t$ and $\lfloor\alpha'\rfloor^t$ cannot be proper subterms of each other.
(The expression $\lfloor\beta\rfloor^t$ represents the term t such that $ev(t) = \beta$. Recall that the logical algebra B is a free algebra, hence for all β there is a unique t such that $ev(t) = \beta$.)

[45] It seems to be characteristic for a logic that it contains at least some equivalent expressions.

[46] It is obvious that for all β and β' either $\mathcal{I}(\beta)\neq\mathcal{I}(\beta')$ or $\mathcal{I}(F_\delta(\beta)) = \mathcal{I}(F_\delta(\beta'))$. That F_δ is non-polynomial is shown in footnote 44 above. An example of such an operator in the logical language of Section 1 is the operator $F : L_{(e,t)}\to L_{(e,t)}$ defined by $F(\text{WALK}) = [[\lambda v_{0,(e,t)}\ v_{0,(e,t)}](\text{WALK})]$, $F([[\lambda v_{0,(e,t)}\ v_{0,(e,t)}](\text{WALK})]) = \text{WALK}$, and $F(\beta) = \beta$ for $\beta\in L_{(e,t)}$ such that $\beta\notin\{\text{WALK}, [[\lambda v_{0,(e,t)}\ v_{0,(e,t)}](\text{WALK})]\}$. (In L it holds for all $\tau\in T$ and $\alpha\in L_\tau$ that $\mathcal{I}([[\lambda v_{0,\tau}\ v_{0,\tau}](\alpha)]) = \mathcal{I}(\alpha)$: note that in all models $m\in\mathcal{M}$ we have that $in_m(\lambda v_{0,\tau}\ v_{0,\tau}) = \{\ \langle a, \{\langle d,d\rangle \mid d\in D_{E,\tau}\}\rangle \mid a\in A\}$.)

Nevertheless, there are good reasons for disregarding operators over the logical algebra B which are only $\mathcal{I}$-functional for *some* semantic algebra $\mathcal{S}$ and epimorphism $\mathcal{I}$ from B to $\mathcal{S}$. For even though formal logical languages usually come with a particular class of models $\mathcal{M}$ which determines a specific semantic algebra $\mathcal{S}$ and epimorphism $\mathcal{I}$ from the logical algebra B to $\mathcal{S}$,[47] this is generally not the class of models in which the translations of (the terms corresponding to) the expressions in the syntactic algebra are interpreted. Most Montague grammar fragments contain a set of so-called *meaning postulates*, which restrict the class $\mathcal{M}$ of all models to a certain subclass.

Meaning postulates are *sentences* of the logical language: expressions of type t which contain no free variables. In a model such expressions denote (a constant function from assignments to) a truth value, viz. 1 ($\{\langle a, 1\rangle \mid a \in A\}$) or 0 ($\{\langle a, 0\rangle \mid a \in A\}$), and a set MP of meaning postulates is intended to reduce the class $\mathcal{M}$ of all models to the subclass $\mathcal{M}^{MP}$ of models in which all meaning postulates in MP are true:[48]

(48) $\mathcal{M}^{MP} = \{m \in \mathcal{M} \mid \text{for all } \varphi \in MP\colon in_m(\varphi) = \{\langle a, 1\rangle \mid a \in A\}\}$

[47]There is, however, some latitude here. Thus in the model theory of languages of typed logic of which the language L introduced in Section 1 is an instance, the models are constructed from one or more basic domains by adding sets of functions between already available domains. Depending on how many functions are added, two kinds of models can be distinguished. In the so-called 'standard' models (the ones defined in Section 1; cf. definition (6) above), the addition clause says that if A and B are domains in the model, then the set B^A of all functions from A to B is also a domain in the model. In the so-called 'generalized' models one may take some subset of the set B^A, where a condition on the construction of models ensures that every object that can be denoted by an expression in the logical language is indeed incorporated into the model. The generalized models were introduced in Henkin (1950) for reducing the non-axiomatizable class of laws of typed logic holding in standard models to an axiomatizable one.

[48]This is only true for languages of extensional typed logic. In the case of intensional logics, where the interpretation of a sentence in a model is (a constant function from assignments to) a function from (sequences of) indices to truth values, the class $\mathcal{M}$ of all models is reduced to the subclass $\mathcal{M}^{MP}$ of models in which all meaning postulates in MP are *valid*, i.e., the class of models in which they denote (the constant function from assignments to) the constant function from (sequences of) indices to the truth value 1.

Normally, meaning postulates are meant to restrict 'the interpretations of the [non-logical] constants of the logic' (Janssen (1986), Part 1, p. 98). This obviously includes the example (51) to be discussed below, but excludes a candidate such as $\exists x \exists y \neg[x = y]$, which restricts the domain of individuals without restricting the interpretation of some non-logical constant.

For all expressions α, the interpretation $\mathcal{I}(\alpha)$ is reduced accordingly:

(49) $\mathcal{I}^{MP}(\alpha) = \{\langle m, in_m(\alpha)\rangle \mid m \in \mathcal{M}^{MP}\}$

The interpretation function $\mathcal{I}^{MP}$ can be construed as an epimorphism from the logical algebra B to the semantic algebra $\mathcal{S}^{MP} = \langle (\mathcal{I}_\tau^{MP})_{\tau\in T}, (\mathcal{G}_\gamma^{MP})_{\gamma\in\Gamma}\rangle$, where

- $\mathcal{I}_\tau^{MP} = \{\mathcal{I}^{MP}(\alpha) \mid \alpha \in L_\tau\}$; and
- if $\pi(\gamma) = \langle\langle\tau_1, \ldots \tau_n\rangle, \tau\rangle$, then $\mathcal{G}_\gamma^{MP} : \mathcal{I}_{\tau_1} \times \ldots \times \mathcal{I}_{\tau_n} \to \mathcal{I}_\tau$, where $\mathcal{G}_\gamma^{MP}(\mathcal{I}^{MP}(\alpha_1), \ldots, \mathcal{I}^{MP}(\alpha_n)) = \mathcal{I}^{MP}(\alpha)$ iff for all $m \in \mathcal{M}^{MP}$: $G_{m,\gamma}(in_m(\alpha_1), \ldots, in_m(\alpha_n)) = in_m(\alpha)$.

It is worth noting that the addition of meaning postulates affects the class of $\mathcal{I}$-functional operators in a fairly inscrutable manner.[49] Consider, for example, the typed logical language L defined in Section 1. With respect to the interpretation homomorphism $\mathcal{I}$ based on the class of models $\mathcal{M}$ defined in (6) through (8), we have that $\mathcal{I}([[\lambda v_{0,\tau}\, v_{0,\tau}](\alpha)]) = \mathcal{I}(\alpha)$ for all $\tau \in T$ and $\alpha \in L_\tau$,[50] so that $\mathcal{I}([[\lambda v_{0,(e,t)}\, v_{0,(e,t)}](\text{WALK})]) = \mathcal{I}(\text{WALK})$, whereas $\mathcal{I}(\text{WALK}) \neq \mathcal{I}(\text{TALK})$, $\mathcal{I}(\text{WALK}) \neq \mathcal{I}(\text{MAN})$ and $\mathcal{I}(\text{TALK}) \neq \mathcal{I}(\text{MAN})$.[51] Therefore, operator F defined in (50) below is not $\mathcal{I}$-functional: $\mathcal{I}([[\lambda v_{0,(e,t)}\, v_{0,(e,t)}](\text{WALK})]) = \mathcal{I}(\text{WALK})$ but $\mathcal{I}(F([[\lambda v_{0,(e,t)} v_{0,(e,t)}](\text{WALK})])) = \mathcal{I}([[\lambda v_{0,(e,t)} v_{0,(e,t)}](\text{WALK})]) = \mathcal{I}(\text{WALK}) \neq \mathcal{I}(\text{TALK}) = \mathcal{I}(F(\text{WALK}))$. On the other hand, operator G in (50) *does* induce a function $G^{\mathcal{I}}$, since if $\mathcal{I}(\alpha) = \mathcal{I}(\alpha')$, then either (*i*) $\mathcal{I}(\alpha)$

The example $\exists x \exists y[\neg[x = y] \wedge [\text{WALK}(x)] \wedge [\text{WALK}(y)]]$ combines both features and shows that the boundary is not always clear. Hence our liberal policy of accepting any expression of type t without free variables as a meaning postulate. Given their function of reducing the class of models, for that matter, it does not even seem essential that meaning postulates are expressions of (or expressible in) the logical language.

(Note, by the way, that – in spite of their name – the distinctness and the update postulate proposed by Groenendijk and Stokhof (1990; cf. Section 6 of Chapter 1 above) are not meaning postulates. They must be considered clauses in the definition of the class of models for the logical language **DIL**, since the 'validity' of these 'postulates' is required for the well-definedness of the denotation of expressions of the form $\{\alpha/d\}\beta$.)

[49] Some results in this area can be distilled from Van Benthem (1980), Section 3.

[50] See footnote 46.

[51] In the model $m \in \mathcal{M}$ such that $D_{E,e} = \{e_1, e_2\}$, $I(\text{WALK}) = \{\langle e_1, 1\rangle, \langle e_2, 0\rangle\}$, $I(\text{TALK}) = \{\langle e_1, 0\rangle, \langle e_2, 1\rangle\}$ and $I(\text{MAN}) = \{\langle e_1, 1\rangle, \langle e_2, 1\rangle\}$ it holds that $in_m(\text{WALK}) = \{\langle a, \{\langle e_1, 1\rangle, \langle e_2, 0\rangle\}\rangle \mid a \in A\}$, $in_m(\text{WALK}) = \{\langle a, \{\langle e_1, 0\rangle, \langle e_2, 1\rangle\}\rangle \mid a \in A\}$ and $in_m(\text{MAN}) = \{\langle a, \{\langle e_1, 1\rangle, \langle e_2, 1\rangle\}\rangle \mid a \in A\}$. Consequently, the functions $in_m(\text{WALK})$, $in_m(\text{TALK})$ and $in_m(\text{MAN})$ are all different.

$= \mathcal{I}(\alpha') \neq \mathcal{I}(\text{WALK})$, hence both $F(\alpha) = \alpha$ and $F(\alpha') = \alpha'$; or (*ii*) $\mathcal{I}(\alpha) = \mathcal{I}(\alpha') = \mathcal{I}(\text{WALK})$, and then $F(\alpha) = \text{MAN} = F(\alpha')$.

(50) $F : L_{(e,t)} \rightarrow L_{(e,t)}$, where $F(\alpha) = \text{TALK}$ if $\alpha = \text{WALK}$, and
$F(\alpha) = \alpha$ if $\alpha \neq \text{WALK}$.
$G : L_{(e,t)} \rightarrow L_{(e,t)}$, where $G(\alpha) = \text{MAN}$ if $\mathcal{I}(\alpha) = \mathcal{I}(\text{WALK})$, and
$G(\alpha) = \alpha$ if $\mathcal{I}(\alpha) \neq \mathcal{I}(\text{WALK})$.

Now, suppose that the class $\mathcal{M}$ of models for L is restricted to the class $\mathcal{M}^{MP}$ by the following singleton set MP of meaning postulates:

(51) $\{\text{WALK} = \text{TALK}\}$

The reduced class $\mathcal{M}^{MP}$ of models comprises the models $m \in \mathcal{M}$ in which the sentence WALK = TALK is true, i.e., those $m \in \mathcal{M}$ such that $in_m(\text{WALK}) = in_m(\text{TALK})$. This has as a consequence that $\mathcal{I}^{MP}(\text{WALK}) = \mathcal{I}^{MP}(\text{TALK})$. But the latter means that the operator F is $\mathcal{I}^{MP}$-functional, since we now have that $\mathcal{I}^{MP}(F(\alpha)) = \mathcal{I}^{MP}(\alpha)$ for all α. On the other hand, the expressions MAN and TALK continue to be non-equivalent under the reduced epimorphism $\mathcal{I}^{MP}$: $\mathcal{I}^{MP}(\text{MAN}) \neq \mathcal{I}^{MP}(\text{TALK})$, and this implies that the operator G *ceases* to induce a function: $\mathcal{I}^{MP}(\text{WALK}) = \mathcal{I}^{MP}(\text{TALK})$, while $\mathcal{I}(\text{WALK}) = \mathcal{I}(\text{WALK})$ but $\mathcal{I}(\text{TALK}) \neq \mathcal{I}(\text{WALK})$, so that $\mathcal{I}^{MP}(G(\text{WALK})) = \mathcal{I}^{MP}(\text{MAN}) \neq \mathcal{I}^{MP}(\text{TALK}) = \mathcal{I}^{MP}(G(\text{TALK}))$.

So, given an initial epimorphism $\mathcal{I}$ and some set MP of meaning postulates, the $\mathcal{I}^{MP}$-functionality of an operator over a logical algebra B cannot be predicted from its $\mathcal{I}$-functionality. Hence it is a safe strategy to allow only those operators over B which are $\mathcal{I}$-functional for *all* epimorphisms $\mathcal{I}$. We have seen above that the class of these universally $\mathcal{I}$-functional operators always includes the polynomial operators over the logical algebra B. Moreover, for the languages of typed logic which are commonly used in Montague grammar fragments there is a complete characterization of this class, because the syntax of such languages constitutes a free algebra B in which each type contains infinitely many generators (viz., the variables of that type). For such algebras it can be shown that the polynomial operators over B actually exhaust the class of operators which are $\mathcal{I}$-functional for all epimorphisms $\mathcal{I}$.[52]

[52] A proof of this result which originates from F. Wiedijk is presented in Appendix 1 of Janssen (1986), Part 1, pp. 189–192 (cf. Van Benthem (1980), footnote 7, for a one-sorted counterpart), where the following claim is proven:

Let $A = \langle (A_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be an algebra. Then $\Pi(A)$, the *polynomial closure* of A, is the following algebra:

(52) $\quad \langle (A_\tau)_{\tau\in T}, \{P^\# \mid P \in \mathrm{POL}^A\}\rangle$

The expression POL^A denotes the set of polynomial symbols over A, while $P^\#$ denotes the polynomial operator determined by the polynomial symbol P (see definition (45) and (46) in Section 3.1 above).

Observe that if $\mathcal{I}$ is an epimorphism from algebra $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ to algebra $C = \langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma}\rangle$, then $\mathcal{I}$ is also an epimorphism from $\Pi(B)$, the polynomial closure of B, to $\Pi(C)$, the polynomial closure of C. This holds on account of the fact that there is a function f from polynomial symbols over B to polynomial symbols over C such that for all $\alpha_1, \ldots, \alpha_n$: $\mathcal{I}(P_\delta^\#(\alpha_1, \ldots, \alpha_n)) = (f(P_\delta))^\#(\mathcal{I}(\alpha_1), \ldots, \mathcal{I}(\alpha_n))$.[53]

When we incorporate the restriction to universally $\mathcal{I}$-functional – i.e.:

Let B be a free algebra $\langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ generated by $(K_\tau)_{\tau\in T}$, where each K_τ is infinite. Let $F_* : L_{\tau_1} \times \ldots \times L_{\tau_n} \to L_\tau$ be an operator such that for every algebra $C = \langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma}\rangle$ and every epimorphism $\mathcal{I}$ from B to C there is a unique operator $G_* : C_{\tau_1} \times \ldots \times C_{\tau_n} \to C_\tau$ such that $\mathcal{I}$ is an epimorphism from $\langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma} \cup \{F_*\}\rangle$ to $\langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma} \cup \{G_*\}\rangle$. Then F_* is polynomially definable.

Of course, an epimorphism $\mathcal{I}$ from $B = \langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ to $C = \langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma}\rangle$ is an epimorphism from $\langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma} \cup \{F_*\}\rangle$ to $\langle (C_\tau)_{\tau\in T}, (G_\gamma)_{\gamma\in\Gamma} \cup \{G_*\}\rangle$ if and only if G_* is a function and $G_* = F_*^{\mathcal{I}}$, the relation $\mathcal{I}$-induced by F_*. So the claim establishes indeed that all universally $\mathcal{I}$-functional operators over an infinitely generated free algebra B are polynomially definable.

Observe that an operator F_* over algebra B is universally $\mathcal{I}$-functional iff the deriver AddOp[$\{F_*\}$] (see Section 3.1 above) is *safe* for B in the sense of definition (41) given above, but that, contrary to what Janssen's motivation for (41) expresses, it is not so much the *uniqueness* as the *existence* of the algebra Q' which is at stake.

If some types in the (free) logical algebra B are finite or finitely generated, then the above result may fail. E.g., if L_τ consists of two expressions α and α', then the non-polynomial operator $F : L_\tau \to L_\tau$ such that $F(\alpha) = \alpha'$ and $F(\alpha') = \alpha$ induces a function $F^{\mathcal{I}}$ for all $\mathcal{I}$. Another example (attributed to W. Peremans in Janssen (1986) and Van Benthem (1980)) concerns the free (one-sorted) algebra $\langle N, S\rangle$, where $N = \{0, S0, SS0, \ldots\}$ and $S : N \to N$ is the (successor) operator defined by $S(n) = Sn$. (This algebra is finitely generated by $\{0\}$: $\langle N, S\rangle = \langle [\{0\}], S\rangle$.) While the (addition) operator $F_* : N \times N \to N$ defined by $F_*(n, 0) = n$ and $F_*(n, S(m)) = S(F_*(n, m))$ is not a polynomal operator over $\langle N, S\rangle$, it does hold that F_* is $\mathcal{I}$-functional for all epimorphisms $\mathcal{I}$.

[53]The function f was defined in Section 3.1. It assigns a polynomial symbol in $\mathrm{POL}^C_{\langle(\tau_1,\ldots\tau_n\rangle,\tau\rangle}$ to each polynomial symbol in $\mathrm{POL}^B_{\langle(\tau_1,\ldots\tau_n\rangle,\tau\rangle}$, viz., $\lfloor \xi^i_\tau \rfloor$ to $\lfloor \xi^i_\tau \rfloor$, $\lfloor \mathcal{I}(c)_\tau \rfloor$ to $\lfloor c_\tau \rfloor$, and $\lfloor \underline{G}_\gamma\, f(p_1) \ldots f(p_k) \rfloor$ to $\lfloor \underline{F}_\gamma\, p_1 \ldots p_k \rfloor$.

polynomial – operators, we eventually arrive at the following situation:

(53)

$$\begin{array}{ll} & T_{A,K} \\ & \downarrow tr \\ B & \Pi(B) \\ \downarrow \mathcal{I} & \downarrow \mathcal{I}^{MP} \\ \mathcal{S} & \Pi(\mathcal{S}^{MP}) \end{array}$$

In (53), B represents a logical algebra $\langle (L_\tau)_{\tau\in T}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ which is interpreted on the basis of a class of models $\mathcal{M}$: the interpretation $\mathcal{I}(\alpha)$ of each logical expression α is a function which associates each $m \in \mathcal{M}$ with $in_m(\alpha)$, the interpretation of α in m. The interpretation function $\mathcal{I}$ is an $(=,=)$-epimorphism from B to a semantic algebra $\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau\in T}, (\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle$, where $\mathcal{I}_\tau = \{\mathcal{I}(\alpha) \mid \alpha \in L_\tau\}$ and $\mathcal{G}_\gamma(\mathcal{I}(\alpha_1),\ldots,\mathcal{I}(\alpha_n)) = \mathcal{I}(\alpha)$ iff for all $m \in \mathcal{M}$: $G_{m,\gamma}(in_m(\alpha_1),\ldots,in_m(\alpha_n)) = in_m(\alpha)$.

In the right-hand side of (53), $T_{A,K}$ represents the term algebra of a syntactic algebra $A = \langle (A_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ generated by $K = (K_c)_{c\in C}$, while $\Pi(B)$ represents the polynomial closure $\langle (L_\tau)_{\tau\in T}, \{P^\# \mid P \in \mathrm{POL}^B\}\rangle$ of the logical algebra B. Every category c of $T_{A,K}$ is assigned a type $\sigma(c)$ of $\Pi(B)$, every operator H_δ^T of $T_{A,K}$ is assigned an operator $P^\#_{\rho(\delta)}$ of $\Pi(B)$, and the translation assignment tr is a (σ,ρ)-homomorphism from $T_{A,K}$ to $\Pi(B)$. MP is a set of meaning postulates which restricts the class $\mathcal{M}$ of models for B to the class $\mathcal{M}^{MP}$ of models in which all postulates in MP are true (or valid), and the restriction $\mathcal{I}^{MP}$ of $\mathcal{I}$ to $\mathcal{M}^{MP}$ is defined by $\mathcal{I}^{MP}(\alpha) = \{\langle m, in_m(\alpha)\rangle \mid m \in \mathcal{M}^{MP}\}$. Since $\mathcal{I}^{MP}$ is an $(=,=)$-epimorphism from B to the semantic algebra $\mathcal{S}^{MP} = \langle (\mathcal{I}_\tau^{MP})_{\tau\in T}, (\mathcal{G}_\gamma^{MP})_{\gamma\in\Gamma}\rangle$, where $\mathcal{I}_\tau^{MP} = \{\mathcal{I}^{MP}(\alpha) \mid \alpha \in L_\tau\}$ and $\mathcal{G}_\gamma^{MP}(\mathcal{I}^{MP}(\alpha_1),\ldots,\mathcal{I}^{MP}(\alpha_n)) = \mathcal{I}^{MP}(\alpha)$ iff for all $m \in \mathcal{M}^{MP}$: $G_{m,\gamma}(in_m(\alpha_1),\ldots,in_m(\alpha_n)) = in_m(\alpha)$, we have that $\mathcal{I}^{MP}$ is also an $(=,=)$-epimorphism from the polynomial closure $\Pi(B)$ of B to $\Pi(\mathcal{S}^{MP})$, the polynomial closure $\langle (\mathcal{I}_\tau^{MP})_{\tau\in T}, \{P^\# \mid P \in \mathrm{POL}^{\mathcal{S}^{MP}}\}\rangle$ of the algebra $\mathcal{S}^{MP}$. As a consequence, the composition $tr \circ \mathcal{I}^{MP}$ of the translation homomorphism tr and the interpretation epimorphism $\mathcal{I}^{MP}$ is a (σ,ρ)-homomorphism from the syntactic term algebra $T_{A,K}$ to the semantic algebra $\Pi(\mathcal{S}^{MP})$.

Summing up, the main advantage of the picture sketched in (53) over the approach outlined in (40) above seems to be that there is no need for a separate process of explicitly deriving algebras. On the one hand, there is a model-theoretically interpreted logic which determines the translation

algebra. On the other hand, there is a grammar fragment consisting of a syntactic algebra, a translation homomorphism to the translation algebra, and a set of meaning postulates. Given the grammar fragment, both the interpretation algebra and the interpretation epimorphism from the translation algebra to the interpretation algebra are induced automatically. This makes the relationship between the grammar of our fragment and the logic that we use in specifying its semantics not only more perspicuous, but also more general: there is no need to readjust our logical tools to every fragment in which we may wish to employ them.

5 Flexible Interpretation

The general idea of flexible semantics as outlined in Chapter 1 is that the expressions of the language do not receive a single interpretation, but a family of systematically related interpretations. Under this conception, scope ambiguities of quantifiers and other operators are different interpretations assigned to the same syntactic analysis of an expression. Montague's strategy of *generalizing to the worst case* (the uniform assignment of the 'highest' type needed for *some* expression in that category to *all* members of a certain syntactic category) is given up and replaced by a more or less reverse strategy: since there is not always a 'highest' type available (i.e., a worst case to generalize to), the functionality of the category-to-type assignment is abandoned. Instead, every expression in the lexicon is given a basic translation of the minimal type available for that expression, and translations of other types are derived from the lexical translation by general rules.

Thus, each *syntactic category* C is not associated with one type, but with a set of types: the *type set* $T(C)$ of category C, which consists of a basic type, $t_B(C)$, together with types which are derived from that type by general rules.

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

For (unambiguous) *lexical* expressions α, the set $Tr_B(\alpha)$ of basic translations of α is a singleton set: $Tr_B(\alpha) = \{tr_B(\alpha)\}$, where $tr_B(\alpha)$ is specified in the lexicon. The other members of $Tr(\alpha)$ are derived from $Tr_B(\alpha)$ by general rules. Not all basic translations are of the basic type: next to such 'best case' translations, there are basic translations of derived types.

For *compound* expressions α_0, the set $Tr_B(\alpha_0)$ of basic translations of

α_0 is a function of (*i*) the translation sets $Tr(\alpha_1), \ldots, Tr(\alpha_n)$ of the parts $\alpha_1, \ldots, \alpha_n$ of α_0 (which sets consist of both the basic and the derived translations of $\alpha_1, \ldots, \alpha_n$); and (*ii*) the translation operation R associated with the syntactic rule that constructs α_0 from its parts $\alpha_1, \ldots, \alpha_n$. That is:

$$Tr_B(\alpha_0) = \{\ R(\alpha'_1, \ldots, \alpha'_n) \mid \alpha'_1 \in Tr(\alpha_1), \ldots, \alpha'_n \in Tr(\alpha_n)\}$$

In conformity with its flexible set-up, the fragment of Chapter 1 employs generalized operations R. For example, the binary operation FA of (intensional) functional application is defined by $\mathrm{FA}(\alpha', \beta') = \alpha'(\hat{}\beta')$ for any pair of expressions α' and β' of intensional logic for which there are types a and b such that α' is of type $((s,a),b)$ and β' is of type a. We noted that this entails a certain partiality with respect to the translation sets $Tr(\alpha_1)$ and $Tr(\alpha_2)$ of the constituent parts α_1 and α_2 of α_0. Only 'mutually fitting' translations $\alpha'_1 \in Tr(\alpha_1)$ and $\alpha'_2 \in Tr(\alpha_2)$ (viz., α'_1 and α'_2 such that α'_1 is of type $((s,a),b)$ and α'_2 is of type a) are capable of producing a translation $\mathrm{FA}(\alpha'_1, \alpha'_2)$ of the compound expression α_0.

Finally, the translation set $Tr(\alpha)$ of a (lexical or compound) expression α is defined inductively as the smallest set which includes the set $Tr_B(\alpha)$ of basic translations of α and is closed under the operations of value raising, argument raising and argument lowering. By way of example, we repeat the definition of argument raising here:[54]

> If $\alpha' \in Tr(\alpha)$ and α' is of type $(\vec{a}, ((s,b), (\vec{c}, d)))$, then
> $\lambda\vec{x}_{\vec{a}}\lambda w_{(s,((s,((s,b),d)),d))}\lambda\vec{y}_{\vec{c}}.{}^{\vee}w(\hat{}\lambda z_{(s,b)}.\alpha'(\vec{x})(z)(\vec{y})) \in Tr(\alpha)$.

Note that the definition of $Tr(\alpha)$ also involves some partiality, since argument raising can only derive translations from translations $\alpha' \in Tr(\alpha)$ which are of type $(\vec{a}, ((s,b), (\vec{c}, d)))$ for some $\vec{a}, b, \vec{c}$ and d.

Clearly, the approach sketched above differs from Montague's treatment of wide scope quantification by means of rules of quantification. The principle

[54] Recall that $\vec{a}$ and $\vec{c}$ are sequences $a_1, \ldots, a_n$ and $c_1, \ldots, c_m$ of types (where $n \geq 0$ and $m \geq 0$); that $(\vec{a}, b)$ and $(\vec{c}, b)$ abbreviate $(a_1, (\ldots (a_n, b) \ldots))$ and $(c_1, (\ldots (c_m, b) \ldots))$; that $\vec{x}_{\vec{a}}$ denotes the sequence $x_1, \ldots, x_n$, where x_i is the i-th variable of type a_i (so that $\vec{x}_{\vec{a}} = v_{1,a_1}, \ldots, v_{n,a_n}$); that $\vec{y}_{\vec{c}}$ denotes the sequence $y_1, \ldots, y_m$, where y_i is the $(n+1+i)$-th variable of type a_i (so that $\vec{y}_{\vec{c}} = v_{n+2,c_1}, \ldots, v_{n+1+m,c_m}$); that z, u and w denote the variables $v_{n+1,(s,b)}$, $v_{n+1,(s,((s,b),d))}$ and $v_{n+1,(s,((s,((s,b),d)),d))}$; and that $\lambda\vec{x}_{\vec{a}}.\tau$, $\lambda\vec{y}_{\vec{c}}.\tau$, $\tau(\vec{x}_{\vec{a}})$ and $\tau(\vec{y}_{\vec{c}})$ abbreviate $\lambda x_1 \ldots \lambda x_n.\tau$, $\lambda y_1 \ldots \lambda y_m.\tau$, $\tau(x_1) \ldots (x_n)$ and $\tau(y_1) \ldots (y_m)$, respectively.

of compositionality entails that in all cases where a non-lexical ambiguity is found that cannot be traced back to the syntactic structure, somehow a syntactic ambiguity needs to be 'forced'. Montague implemented this by creating derivational ambiguities: whenever an ambiguity cannot be accounted for in terms of different syntactic structures (*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.

In the flexible grammar outlined above such ambiguities do not have syntactic repercussions. Instead, they are accounted for by associating syntactic objects with *sets* of interpretations.

On the one hand, this approach apparently yields a grammar which is 'more compositional' than grammars employing rules of quantification. In representing scope ambiguities no use is made of 'artificial' alternative ways of construction. All syntactic operations of the fragment respect the 'intuitive' syntactic structure, so that constituent expressions are real *parts*.

On the other hand, although the interpretations of a compound expression are determined by the interpretations of its constituent parts and the semantic operation associated with the compounding syntactic operation, we can no longer speak of *the* interpretation of a constituent part. All expressions have sets of interpretations, of which only some members are able to join. In this survival of the fitting, the operations of value raising, argument raising and argument lowering enable expressions to adapt themselves to their semantic environment. Hence at first glance, 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 Vereinzelung gefragt werden [...].
> (Frege (1884), p. XXII).[55]

Under a weak interpretation, 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'[56] –, we must study occurrences of that expression in the context of sentences.

[55] 'The meaning of words can only be found in the context of sentences, not when they are studied in isolation.'

[56] Dummett arrives at this weak interpretation in his discussion of the status of the principle of contextuality in Frege's work: '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

The weak interpretation can be opposed to a strong interpretation, which is suggested by the following statement:

> Nur im Zusammenhange eines Satzes bedeuten die Wörter etwas. (Frege (1884), p. 73).[57]

Under its strong interpretation, the principle of contextuality explicitly denies that expressions have independent meanings: in different sentences an expression contributes in different ways to the truth conditions of the whole in which it occurs.[58]

In the flexible grammar this seems to be the case. For instance, the semantic contribution of the expression *finds* to the truth conditions of the sentence *John finds Mary* can be formalized as (54), which is the basic translation of *finds*. Besides, this transitive verb contributes in two different ways to the truth conditions of the sentence *Every man finds a woman*, viz., (55) and (56), which non-equivalently result in a wide scope-*every* and a wide scope-*a* reading:

(54) $\lambda y \lambda x.\text{FIND}(^{\vee}y)(^{\vee}x)$

(55) $\lambda T_1 \lambda T_2.^{\vee}T_2(^{\wedge}\lambda y.^{\vee}T_1(^{\wedge}\lambda x.\text{FIND}(^{\vee}x)(^{\vee}y)))$

(56) $\lambda T_1 \lambda T_2.^{\vee}T_1(^{\wedge}\lambda x.^{\vee}T_2(^{\wedge}\lambda y.\text{FIND}(^{\vee}x)(^{\vee}y)))$

Note, moreover, that the basic translation (54) cannot be used for interpeting the sentence *Every man finds a woman*, and that the combination of the derived translations (55) and (56) with the derived translations $\lambda P.^{\vee}P(^{\wedge}m)$ and $\lambda P.^{\vee}P(^{\wedge}j)$ of *Mary* and *John* leads to equivalent translations for the sentence *John finds Mary*.

---

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).

The fact that within the grammar of PTQ all expressions are assigned interpretations of a relational type $(\vec{a}, t)$ illustrates the latter point: the interpretation of expressions is identified with their contribution to the truth conditions of the sentences in which they occur.

[57]'It is only in the context of a sentence that words have a meaning.'

[58]As Janssen notes, this version of contextuality 'seems to be in conflict with compositionality: the principle of compositionality requires that words in isolation have a meaning, since otherwise there is nothing from which the meaning of a compound expression can be built' (Janssen (1986), Part 1, p. 6).

In spite of these observations, however, it would be wrong to conclude that the grammar exemplifies the strong version of the principle of contextuality. For even though an expression may contribute in different ways to the truth conditions of the various sentences in which it occurs, these different contributions do all have a 'greatest common divisor', viz., the 'product' of the interpretations expressed by the basic translations of its lexical parts. Whenever a (basic or derived) translation of some expression is combined with translations of other expressions, the basic translations of its lexical subexpressions are present as well, since they make up the very building blocks of that translation. This becomes apparent if the derivation of translations is explicitly incorporated into the grammar. The result of this incorporation is a completely equivalent fragment which is downright compositional.

The procedure is straightforward: the operations of value raising, argument raising and argument lowering are recasted as unary grammatical operations which, when applied to an expression with a translation of a certain type, return that expression with a translation of a different type.

This means that we need more fine-grained syntactic sorts, e.g., C_a, where C is a syntactic category and a is a type. (Given this, the category-to-type assignment σ can be kept very simple: $\sigma(C_a) = a$.)

(57) C_a is a *syntactic sort* iff $C \in$ {S,T,CN,IV,Det,TV,PV} and $a \in T$, where T, the set of *types*, is the smallest set such that
(i) $e \in T$ and $t \in T$;
(ii) if $a \in T$ and $b \in T$, then $(a, b) \in T$; and
(iii) if $b \in T$, then $(s, b) \in T$.

The lexicon of the fragment is given in (58) (where the types $((s,e),t)$ and $((s,((s,e),t)),t)$ are abbreviated as e^1 and e^2, respectively):[59]

(58)

α	CATEGORY OF α	*lextr*(α)
price	$\mathrm{CN}_{((s,e),t)}$	PRICE
man	$\mathrm{CN}_{((s,e),t)}$	$\lambda x.[\mathrm{MAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$
woman	$\mathrm{CN}_{((s,e),t)}$	$\lambda x.[\mathrm{WOMAN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$
unicorn	$\mathrm{CN}_{((s,e),t)}$	$\lambda x.[\mathrm{UNICORN}(^{\vee}x) \wedge \exists u(x = {}^{\wedge}u)]$

[59] Observe that the labelled bracketings $[_c\ \alpha]$ of the original fragment have been replaced by their expressions α (cf. Section 1.3 of Chapter 1). By using labelled bracketings the fragment could be kept syntactically unambiguous, so that the necessity of introducing and interpreting its term algebra could be circumvented. The use of expressions has the advantage of reducing the number of subscripts (which is large enough as it is). Apart from this, nothing further hinges on the choice.

is missing	$\mathrm{IV}_{((s,e^2),t)}$	BE-MISSING
walks	$\mathrm{IV}_{((s,e),t)}$	$\lambda x.\mathrm{WALK}(^\vee x)$
rises	$\mathrm{IV}_{((s,e),t)}$	RISE
John	T_e	j
Mary	T_e	m
a	$\mathrm{Det}_{((s,e^1),e^2)}$	$\lambda P\lambda Q.\exists x[^\vee P(x) \wedge {}^\vee Q(x)]$
the	$\mathrm{Det}_{((s,e^1),e^2)}$	$\lambda P\lambda Q.\exists x[\forall y[^\vee P(y) \leftrightarrow x = y] \wedge {}^\vee Q(x)]$
every	$\mathrm{Det}_{((s,e^1),e^2)}$	$\lambda P\lambda Q.\forall x[^\vee P(x) \rightarrow {}^\vee Q(x)]$
one	$\mathrm{Det}_{((s,e^1),e^2)}$	$\lambda P\lambda Q.\exists x\forall y[[^\vee P(y) \wedge {}^\vee Q(y)] \leftrightarrow x = y]$
seeks	$\mathrm{TV}_{((s,e^2),e^1)}$	$\lambda T\lambda x.\mathrm{SEEK}(T)(^\vee x)$
finds	$\mathrm{TV}_{((s,e),e^1)}$	$\lambda y\lambda x.\mathrm{FIND}(^\vee y)(^\vee x)$
claims that	$\mathrm{PV}_{((s,t),e^1)}$	$\lambda p\lambda x.\mathrm{CLAIM}(p)(^\vee x)$
believes that	$\mathrm{PV}_{((s,t),e^1)}$	$\lambda p\lambda x.\mathrm{BELIEVE}(p)(^\vee x)$

Next, assume that b and d are types, that $\vec{a}$ and $\vec{c}$ are finite sequences of types, that $A \in \{\mathrm{S,IV,TV,CN}\}$ and that $B \in \{\mathrm{S,IV,T,TV,CN}\}$. For each syntactic sort C_a, the set E_{C_a} of English expressions of sort C_a is defined as the smallest set such that:

(59) if α of sort C_a occurs in (58), then $\alpha \in E_{C_a}$;
if $\alpha \in E_{\mathrm{T}_{((s,b),d)}}$ and $\beta \in E_{\mathrm{IV}_b}$, then $H_{1:b:d}(\alpha,\beta) \in E_{\mathrm{S}_d}$;
if $\alpha \in E_{\mathrm{TV}_{((s,b),d)}}$ and $\beta \in E_{\mathrm{T}_b}$, then $H_{2:b:d}(\alpha,\beta) \in E_{\mathrm{IV}_d}$;
if $\alpha \in E_{\mathrm{Det}_{((s,b),d)}}$ and $\beta \in E_{\mathrm{CN}_b}$, then $H_{3:b:d}(\alpha,\beta) \in E_{\mathrm{T}_d}$;
if $\alpha \in E_{\mathrm{PV}_{((s,b),d)}}$ and $\beta \in E_{\mathrm{S}_b}$, then $H_{4:b:d}(\alpha,\beta) \in E_{\mathrm{IV}_d}$;
if $\alpha \in E_{A_{(\vec{a},t)}}$ and $\beta \in E_{A_{(\vec{a},t)}}$, then $H_{5:A:\vec{a}}(\alpha,\beta) \in E_{A_{(\vec{a},t)}}$;
if $\alpha \in E_{B_{(\vec{a},t)}}$ and $\beta \in E_{B_{(\vec{a},t)}}$, then $H_{6:B:\vec{a}}(\alpha,\beta) \in E_{B_{(\vec{a},t)}}$;
if $\alpha \in E_{C_{(\vec{a},b)}}$, then $H_{7:C:\vec{a}:b:d}(\alpha) \in E_{C_{(\vec{a},((s,((s,b),d)),d))}}$;
if $\alpha \in E_{C_{(\vec{a},((s,b),(\vec{c},d)))}}$, then $H_{8:C:\vec{a}:b:\vec{c}:d}(\alpha) \in E_{C_{(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d)))}}$;
if $\alpha \in E_{C_{(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d)))}}$, then $H_{9:C:\vec{a}:b:\vec{c}:d}(\alpha) \in E_{C_{(\vec{a},((s,b),(\vec{c},d)))}}$.

The syntactic operators mentioned in (59) are defined in (60):

(60) $H_{1:b:d} : E_{\mathrm{T}_{((s,b),d)}} \times E_{\mathrm{IV}_b} \rightarrow E_{\mathrm{S}_d}$, where $H_{1:b:d}(\alpha,\beta) = \alpha\,\beta$;
$H_{2:b:d} : E_{\mathrm{TV}_{((s,b),d)}} \times E_{\mathrm{T}_b} \rightarrow E_{\mathrm{IV}_d}$, where $H_{2:b:d}(\alpha,\beta) = \alpha\,\beta$;
$H_{3:b:d} : E_{\mathrm{Det}_{((s,b),d)}} \times E_{\mathrm{CN}_b} \rightarrow E_{\mathrm{T}_d}$, where $H_{3:b:d}(\alpha,\beta) = \alpha\,\beta$;
$H_{4:b:d} : E_{\mathrm{PV}_{((s,b),d)}} \times E_{\mathrm{S}_b} \rightarrow E_{\mathrm{IV}_d}$, where $H_{4:b:d}(\alpha,\beta) = \alpha\,\beta$;
$H_{5:A:\vec{a}} : E_{A_{(\vec{a},t)}} \times E_{A_{(\vec{a},t)}} \rightarrow E_{A_{(\vec{a},t)}}$, where $H_{5:A:\vec{a}}(\alpha,\beta) = \alpha\ and\ \beta$;
$H_{6:B:\vec{a}} : E_{B_{(\vec{a},t)}} \times E_{B_{(\vec{a},t)}} \rightarrow E_{B_{(\vec{a},t)}}$, where $H_{6:B:\vec{a}}(\alpha,\beta) = \alpha\ or\ \beta$;

$H_{7:C:\vec{a}:b:d} : E_{C_{(\vec{a},b)}} \rightarrow E_{C_{(\vec{a},((s,((s,b),d)),d))}}$,
where $H_{7:C:\vec{a}:b:d}(\alpha) = \alpha$;
$H_{8:C:\vec{a}:b:\vec{c}:d} : E_{C_{(\vec{a},((s,b),(\vec{c},d)))}} \rightarrow E_{C_{(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d)))}}$,
where $H_{8:C:\vec{a}:b:\vec{c}:d}(\alpha) = \alpha$;
$H_{9:C:\vec{a}:b:\vec{c}:d} : E_{C_{(\vec{a},((s,((s,((s,b),d)),d)),(\vec{c},d)))}} \rightarrow E_{C_{(\vec{a},((s,b),(\vec{c},d)))}}$,
where $H_{9:C:\vec{a}:b:\vec{c}:d}(\alpha) = \alpha$.

The binary operations $H_{1:b:d}$ through $H_{4:b:d}$ replace the rules of application of the original fragment; the respective operations $H_{5:A:\vec{a}}$ and $H_{6:B:\vec{a}}$ collapse its four rules of conjunction and its five rules of disjunction; and the unary operations $H_{7:C:\vec{a}:b:d}$, $H_{8:C:\vec{a}:b:\vec{c}:d}$ and $H_{9:C:\vec{a}:b:\vec{c}:d}$ take the place of value raising, argument raising and argument lowering, respectively. Accordingly, the syntactic terms which correspond to the expressions defined by the syntactic operations are assigned the following translations:

(61)
$tr(\lfloor \alpha_{C_a} \rfloor) = lextr(\alpha)$ for α of category C_a in (58);
$tr(\lfloor \underline{H}_{1:b:d}\, \alpha\beta \rfloor) = [tr(\alpha)](\hat{}[tr(\beta)])$;
$tr(\lfloor \underline{H}_{2:b:d}\, \alpha\beta \rfloor) = [tr(\alpha)](\hat{}[tr(\beta)])$;
$tr(\lfloor \underline{H}_{3:b:d}\, \alpha\beta \rfloor) = [tr(\alpha)](\hat{}[tr(\beta)])$;
$tr(\lfloor \underline{H}_{4:b:d}\, \alpha\beta \rfloor) = [tr(\alpha)](\hat{}[tr(\beta)])$;
$tr(\lfloor \underline{H}_{5:A:\vec{a}}\, \alpha\beta \rfloor) = \lambda\vec{x}.[[tr(\alpha)](\vec{x}) \wedge [tr(\beta)](\vec{x})]$;
$tr(\lfloor \underline{H}_{6:B:\vec{a}}\, \alpha\beta \rfloor) = \lambda\vec{x}.[[tr(\alpha)](\vec{x}) \vee [tr(\beta)](\vec{x})]$;
$tr(\lfloor \underline{H}_{7:C:\vec{a}:b:d}\, \alpha \rfloor) = \lambda\vec{x}\lambda u.{}^{\vee}u(\hat{}[[tr(\alpha)](\vec{x})])$;
$tr(\lfloor \underline{H}_{8:C:\vec{a}:b:\vec{c}:d}\, \alpha \rfloor) = \lambda\vec{x}\lambda w\lambda\vec{y}.{}^{\vee}w(\hat{}\lambda z.[tr(\alpha)](\vec{x})(z)(\vec{y}))$;
$tr(\lfloor \underline{H}_{9:C:\vec{a}:b:\vec{c}:d}\, \alpha \rfloor) = \lambda\vec{x}\lambda z\lambda\vec{y}.[tr(\alpha)](\vec{x})(\hat{}\lambda u.{}^{\vee}u(z))(\vec{y})$.

These translations are expressions of a language $\mathbf{IL} = (L_\tau)_{\tau \in T}$ of intensional typed logic, where T is the set of types defined in (57). For every type $\tau \in T$, there are two sets of basic expressions: a set V_τ of variables of type τ, and a set C_τ of constants of type τ:

(62)
$V_\tau = \{v_{i,\tau} \mid i \in \mathbb{N}\}$;
$C_e = \{m, j\}$, $C_{(e,(e,t))} = \{\text{FIND}\}$, $C_{((s,((s,(s,e),t)),t)),t)} = \{\text{BE-MISSING}\}$,
$C_{(e,t)} = \{\text{MAN}, \text{WOMAN}, \text{UNICORN}, \text{WALK}\}$, $C_{((s,e),t)} = \{\text{PRICE}, \text{RISE}\}$,
$C_{((s,((s,((s,e),t)),t)),(e,t))} = \{\text{SEEK}\}$, $C_{((s,t)(e,t))} = \{\text{CLAIM}, \text{BELIEVE}\}$,
and $C_\tau = \emptyset$ otherwise.

For every type $\tau \in T$, the set L_τ is the smallest set such that:

(63)
$V_\tau \subseteq L_\tau$ and $C_\tau \subseteq L_\tau$;
if $\varphi \in L_t$, then $F_1(\varphi) \in L_t$;
if $\varphi \in L_t$ and $\psi \in L_t$, then $F_2(\varphi, \psi) \in L_t$;

if $\varphi \in L_t$, then $F_{3:\tau:i}(\varphi) \in L_t$;
if $\alpha \in L_{(\sigma,\tau)}$ and $\beta \in L_\sigma$, then $F_{4:\sigma:\tau}(\alpha, \beta) \in L_\tau$;
if $\alpha \in L_\tau$, then $F_{5:\sigma:\tau:i}(\alpha) \in L_{(\sigma,\tau)}$;
if $\alpha \in L_\tau$ and $\beta \in L_\tau$, then $F_{6:\tau}(\alpha, \beta) \in L_t$;
if $\alpha \in L_\tau$, then $F_{7:\tau}(\alpha) \in L_{(s,\tau)}$;
if $\alpha \in L_{(s,\tau)}$, then $F_{8:\tau}(\alpha) \in L_\tau$.

The logical operations in (63) are defined in (64) for $\sigma, \tau \in T$ and $i \in \mathbb{N}$:

(64) $F_1 : L_t \to L_t$, where $F_1(\varphi) = \neg\, \varphi$;
$F_2 : L_t \times L_t \to L_t$, where $F_2(\varphi, \psi) = [\varphi \wedge \psi]$;
$F_{3:\tau:i} : L_t \to L_t$, where $F_{3:\tau:i}(\varphi) = \forall v_{i,\tau}\, \varphi$;
$F_{4:\sigma:\tau} : L_{(\sigma,\tau)} \times L_\sigma \to L_\tau$, where $F_{4:\sigma:\tau}(\alpha, \beta) = [[\alpha](\beta)]$;
$F_{5:\sigma:\tau:i} : L_\tau \to L_{(\sigma,\tau)}$, where $F_{5:\sigma:\tau:i}(\alpha) = \lambda v_{i,\sigma}\, \alpha$;
$F_{6:\tau} : L_\tau \times L_\tau \to L_t$, where $F_{6:\tau}(\alpha, \beta) = (\alpha = \beta)$;
$F_{7:\tau} : L_\tau \to L_{(s,\tau)}$, where $F_{7:\tau}(\alpha) = {}^\wedge\alpha$;
$F_{8:\tau} : L_{(s,\tau)} \to L_\tau$, where $F_{8:\tau}(\alpha) = {}^\vee\alpha$.

As usual, we let the expressions $[\varphi \vee \psi]$, $[\varphi \to \psi]$ and $\exists v_{i,\tau}\, \varphi$ abbreviate $\neg\,[\neg\varphi \wedge \neg\psi]$, $\neg\,[\varphi \wedge \neg\psi]$ and $\neg\, \forall v_{i,\tau}\, \neg\, \varphi$, respectively.[60]

Now, it can be noted that all operators in the syntactic term algebra correspond to polynomial operators over the logic. This is obvious for $H^T_{1:b:d}$ through $H^T_{4:b:d}$. Since $H_{i:b:d}(\alpha, \beta) = \alpha({}^\wedge\beta)$ for $i \in \{1, 2, 3, 4\}$, the logical operators corresponding to these rules of intensional functional application are defined by polynomial symbols $\lfloor \underline{F}_{4:(s,b):d} \lfloor \xi^1_{((s,b),d)} \rfloor \lfloor \underline{F}_{7:b} \lfloor \xi^2_b \rfloor \rfloor \rfloor$.

But it also holds for the operators $H^T_{5:A:\vec{a}}$ through $H^T_{9:C:\vec{a}:b:\vec{c}:d}$. In footnote 54 it was recalled that for each sequence $\vec{a} = a_1, \ldots, a_n$ and $\vec{c} = c_1, \ldots, c_m$

[60] For the necessary rigour we note here that a model-theoretic interpretation of **IL** is based on a non-empty set E of individuals and a non-empty set W of possible worlds which determine a domain $D_{E,W,\tau}$ for every $\tau \in T$, viz.: $D_{E,W,e} = E$, $D_{E,W,t} = \{0, 1\}$, $D_{E,W,(\sigma,\tau)} = D^{D_{E,W,\sigma}}_{E,W,\tau}$, $D_{E,(s,\tau)} = D^W_{E,W,\tau}$; that a frame is a family $(D_{E,W,\tau})_{\tau \in T}$; that constants $c \in C_\tau$ are assigned an interpretation $I(c) \in D^W_{E,W,\tau}$ (a function from possible worlds to objects in $D_{E,W,\tau}$!); that variable assignments $a \in A$ assign an object $a(v_{i,\tau}) \in D_{E,W,\tau}$ to each variable $v_{i,\tau}$; that a *model* m consists of a frame and a constant interpretation: $m = \langle (D_{E,W,\tau})_{\tau \in T}, I \rangle$; and that the interpretation $in_m(\alpha)$ in m of a well-formed expression α of type τ is the following function from A and W to $D_{E,W,\tau}$:
if $\alpha \in V_\tau$, then $in_m(\alpha) = \{\langle \langle a, w \rangle, a(\alpha) \rangle \mid a \in A \text{ and } w \in W\}$;
if $\alpha \in C_\tau$, then $in_m(\alpha) = \{\langle \langle a, w \rangle, I(\alpha)(w) \rangle \mid a \in A \text{ and } w \in W\}$;
$in_m(F_1(\varphi)) = G_{m,1}(in_m(\varphi))$; $in_m(F_2(\varphi, \psi)) = G_{m,2}(in_m(\varphi), in_m(\psi))$;
$in_m(F_{3:\tau:i}(\varphi)) = G_{m,3:\tau:i}(in_m(\varphi))$; $in_m(F_{4:\sigma:\tau}(\alpha, \beta)) = G_{m,4:\sigma:\tau}(in_m(\alpha), in_m(\beta))$;
$in_m(F_{5:\sigma:\tau:i}(\alpha)) = G_{m,5:\sigma:\tau:i}(in_m(\alpha))$; $in_m(F_{6:\tau}(\alpha, \beta)) = G_{m,6:\tau}(in_m(\alpha), in_m(\beta))$;
$in_m(F_{7:\tau}(\alpha)) = G_{m,7:\tau}(in_m(\alpha))$; $in_m(F_{8:\tau}(\alpha)) = G_{m,8:\tau}(in_m(\alpha))$; where

of types, the symbols $\vec{x}_{\vec{a}}$ and $\vec{y}_{\vec{c}}$ denote *specific* sequences of variables, viz., $v_{1,a_1}, \ldots, v_{n,a_n}$ and $\vec{y}_{\vec{c}} = v_{n+2,c_1}, \ldots, v_{n+1+m,c_m}$, respectively, and that the respective symbols z, u and w denote the variables $v_{n+1,(s,b)}$, $v_{n+1,(s,((s,b),d))}$ and $v_{n+1,(s,((s,((s,b),d)),d))}$. Therefore, since

$$\begin{aligned} H_{5:A:\vec{a}}(\alpha,\beta) &= \lambda\vec{x}.[\alpha(\vec{x}) \wedge \beta(\vec{x})], \\ H_{6:B:\vec{a}}(\alpha,\beta) &= \lambda\vec{x}.[\alpha(\vec{x}) \vee \beta(\vec{x})], \text{ and} \\ H_{7:C:\vec{a}:b:d}(\alpha) &= \lambda\vec{x}\lambda u.^{\vee}u(^{\wedge}[\alpha(\vec{x})]), \end{aligned}$$

the logical operators corresponding to $H^T_{5:A:\vec{a}}$, $H^T_{6:B:\vec{a}}$ and $H^T_{7:C:\vec{a}:b:d}$ are defined by the respective polynomial symbols[61]

$$\begin{aligned} &\lfloor P_1 \ldots \lfloor P_n \lfloor \underline{F}_2 \mathrm{A}\, \mathrm{B} \rfloor \rfloor \cdots \rfloor, \\ &\lfloor P_1 \ldots \lfloor P_n \lfloor \underline{F}_1 \lfloor \underline{F}_2 \lfloor \underline{F}_1 \mathrm{A} \rfloor \lfloor \underline{F}_1 \mathrm{B} \rfloor \rfloor \rfloor \rfloor \cdots \rfloor, \text{ and} \\ &\lfloor Q_1 \ldots \lfloor Q_n \lfloor R \lfloor \underline{F}_{4:(s,b):d} \lfloor \underline{F}_{7:((s,b),d)} \lfloor v_{n+1,(s,((s,b),d))} \rfloor \rfloor \lfloor \underline{F}_{8:b} \mathrm{C} \rfloor \rfloor \rfloor \rfloor \cdots \rfloor, \end{aligned}$$

where $P_i = \underline{F}_{5:a_i:(a_{i+1},(\ldots(a_n,t)\ldots)):i}$ and $Q_i = \underline{F}_{5:a_i:(a_{i+1},(\ldots(a_n,((s,((s,b),d)),d))\ldots)):i}$ for all i such that $1 \leq i \leq n$; $R = \underline{F}_{5:(s,((s,b),d)):d:n+1}$; and

---

$G_{m,1} : D^{A\times W}_{E,W,t} \to D^{A\times W}_{E,W,t}$ and $G_{m,1}(\delta) =$
$\{\langle\langle a,w\rangle,1\rangle \mid \langle\langle a,w\rangle,0\rangle \in \delta\} \cup$
$\{\langle\langle a,w\rangle,0\rangle \mid \langle\langle a,w\rangle,1\rangle \in \delta\}$;
$G_{m,2} : D^{A\times W}_{E,W,t} \times D^{A\times W}_{E,W,t} \to D^{A\times W}_{E,W,t}$ and $G_{m,2}(\delta,\varepsilon) =$
$\{\langle\langle a,w\rangle,1\rangle \mid \langle\langle a,w\rangle,1\rangle \in \delta$ and $\langle\langle a,w\rangle,1\rangle \in \varepsilon\} \cup$
$\{\langle\langle a,w\rangle,0\rangle \mid \langle\langle a,w\rangle,0\rangle \in \delta$ or $\langle\langle a,w\rangle,0\rangle \in \varepsilon\}$;
$G_{m,3:\tau:i} : D^{A\times W}_{E,W,t} \to D^{A\times W}_{E,W,t}$ and $G_{m,3:\tau:i}(\delta) =$
$\{\langle\langle a,w\rangle,1\rangle \mid$ for every $d \in D_{E,W,\tau}$: $\langle\langle a[v_{i,\tau}/d],w\rangle,1\rangle \in \delta\} \cup$
$\{\langle\langle a,w\rangle,0\rangle \mid$ for some $d \in D_{E,W,\tau}$: $\langle\langle a[v_{i,\tau}/d],w\rangle,0\rangle \in \delta\}$;
$G_{m,4:\sigma:\tau} : D^{A\times W}_{E,W,(\sigma,\tau)} \times D^{A\times W}_{E,W,\sigma} \to D^{A\times W}_{E,W,\tau}$ and $G_{m,4:\sigma:\tau}(\delta,\varepsilon) =$
$\{\langle\langle a,w\rangle,f(d)\rangle \mid \langle\langle a,w\rangle,f\rangle \in \delta$ and $\langle\langle a,w\rangle,d\rangle \in \varepsilon\}$;
$G_{m,5:\sigma:\tau:i} : D^{A\times W}_{E,W,\tau} \to D^{A\times W}_{E,W,(\sigma,\tau)}$ and $G_{m,5:\sigma:\tau:i}(\delta) =$
$\{\langle\langle a,w\rangle,f\rangle \mid$ for every $d \in D_{E,W,\sigma}$: $\langle\langle a[v_{i,\sigma}/d],w\rangle,f(d)\rangle \in \delta\}$;
$G_{m,6:\tau} : D^{A\times W}_{E,W,\tau} \times D^{A\times W}_{E,W,\tau} \to D^{A\times W}_{E,W,t}$ and $G_{m,6:\tau}(\delta,\varepsilon) =$
$\{\langle\langle a,w\rangle,1\rangle \mid \langle\langle a,w\rangle,d\rangle \in \delta$ and $\langle\langle a,w\rangle,d'\rangle \in \varepsilon$ and $d = d'\} \cup$
$\{\langle\langle a,w\rangle,0\rangle \mid \langle\langle a,w\rangle,d\rangle \in \delta$ and $\langle\langle a,w\rangle,d'\rangle \in \varepsilon$ and $d \neq d'\}$;
$G_{m,7:\tau} : D^{A\times W}_{E,W,\tau} \to D^{A\times W}_{E,W,(s,\tau)}$ and $G_{m,7:\tau}(\delta) =$
$\{\langle\langle a,w\rangle,f\rangle \mid$ for every $w' \in W$: $\langle\langle a,w'\rangle,f(w')\rangle \in \delta\}$;
$G_{m,8:\tau} : D^{A\times W}_{E,W,(s,\tau)} \to D^{A\times W}_{E,W,\tau}$ and $G_{m,8:\tau}(\delta) =$
$\{\langle\langle a,w\rangle,f(w)\rangle \mid \langle\langle a,w\rangle,f\rangle \in \delta\}$.

[61] In the second polynomial symbol, the definition of $[\varphi \vee \psi]$ in terms of $\neg[\neg\varphi \wedge \neg\psi]$ has been exploited. Note that, very strictly speaking, the third polynomial symbol should contain $\lfloor v_{n+1,(s,((s,b),d)),(s,((s,b),d))} \rfloor$ instead of $\lfloor v_{n+1,(s,((s,b),d))} \rfloor$, since a logical variable $v_{n+1,\tau}$ determines a polynomial constant $v_{n+1,\tau,\tau}$ which determines a polynomial symbol $\lfloor v_{n+1,\tau,\tau} \rfloor$.

$$A = \lfloor R_n \lfloor \ldots \lfloor R_1 \lfloor \xi^1_{(\vec{a},t)} \rfloor \lfloor v_{1,a_1,a_1} \rfloor \rfloor \ldots \rfloor \lfloor v_{n,a_n,a_n} \rfloor \rfloor,$$
$$B = \lfloor R_n \lfloor \ldots \lfloor R_1 \lfloor \xi^2_{(\vec{a},t)} \rfloor \lfloor v_{1,a_1,a_1} \rfloor \rfloor \ldots \rfloor \lfloor v_{n,a_n,a_n} \rfloor \rfloor, \text{ and}$$
$$C = \lfloor S_n \lfloor \ldots \lfloor S_1 \lfloor \xi^1_{(\vec{a},b)} \rfloor \lfloor v_{1,a_1,a_1} \rfloor \rfloor \ldots \rfloor \lfloor v_{n,a_n,a_n} \rfloor \rfloor,$$

where $R_i = \underline{F}_{4:a_i:(a_{i+1},(\ldots(a_n,t)\ldots))}$ and $S_i = \underline{F}_{4:a_i:(a_{i+1},(\ldots(a_n,b)\ldots))}$ for $1 \leq i \leq n$. Analogously, it can be shown that the logical operators corresponding to $H^T_{8:C:\vec{a}:b:\vec{c}:d}$ and $H^T_{9:C:\vec{a}:b:\vec{c}:d}$ are defined by polynomial symbols. Fear of overfull \hboxes, however, restrained us from actually accomplishing this task.

Observe, finally, that the above compositional reformulation of the flexible grammar bears out the idea that this grammar is 'more compositional' than fragments which employ rules of quantification because no use is made of 'artificial' alternative ways of construction and all syntactic operations of the fragment respect the 'intuitive' syntactic structure. This appears from the fact that every syntactic operation in (60) either concatenates its arguments (cf. $H_{1:b:d}$ through $H_{6:A:\vec{a}}$; note that $H_{5:A:\vec{a}}$ and $H_{6:A:\vec{a}}$ also introduce syncategorematic expressions) or returns its sole argument (this holds for $H_{7:C:\vec{a}:b:d}$ through $H_{9:C:\vec{a}:b:\vec{c}:d}$). Hence the grammar observes the 'most intuitive interpretation of the principle of compositionality, which says that the parts of a compound expression have to be *visible* parts of the compound expression' (Janssen, (1986), Part 1, p. 65–66).

Chapter 3

Lambek Semantics

... ὁδὸς ἄνω κάτω μία καὶ ὠυτή.
... [the] way up [and] down [is] one and [the] same.

Heraclitus, *Diels–Kranz 22 B 60*

When Lambek (1958) introduced his syntactic calculus, he showed that it is equivalent to a sequent axiomatization, the Lambek-Gentzen sequent calculus (henceforth: **L**). The calculus **L** defines a general notion of derivability in the following sense: an expression consisting of the lexical items $e_1, \ldots, e_n$ of respective categories $c_1, \ldots, c_n$ is parsed as belonging to a certain category c if and only if the statement '$c_1, \ldots, c_n$ is a c' (written as a so-called *sequent* $c_1, \ldots, c_n \vdash c$) can be derived as a theorem of the system. Thus, grammatical derivations are reduced to logical deductions, giving rise to the slogan 'parsing as deduction'. The calculus **L** contains, among other rules, the so-called *Cut* rule. By proving that the set of theorems of **L** is not increased by adding *Cut*, Lambek established the decidability of **L**: for an arbitrary sequent the proof procedure is guaranteed to answer the question whether the sequent is valid after a finite number of steps. The proof of this fact is constructive: Lambek defined an algorithm which enables one to eliminate all occurrences of *Cut* in proofs.

The semantic interpretation procedure of Moortgat (1988) for product-free **L** is a sequent implementation of Van Benthem's term construction algorithm for the natural deduction formulation of **L** (1986, Chapter 7; 1991). In this semantics, the interpretations of a grammatical expression (derivable sequent) are directly determined by the proofs of its validity in the syntactic calculus. The present chapter provides an alternative formulation of the Van Benthem/Moortgat semantics for **L** (which is introduced in Section 1) and uses it to prove the following results. Section 2 contains a straightforward proof of a semantic version of Lambek's *Cut* elimination theorem which entails that **L** is semantically decidable as well: the result of applying Lambek's *Cut* elimination algorithm is a derivation which is semantically equivalent to the original derivation. Section 3 is devoted to further normalization of

L, resulting in a sequent calculus **L*** which is put to use in the subsequent sections. First, we show in Section 4 that **L*** provides a solution to the so-called 'spurious ambiguity problem' – the problem that different proofs of a given sequent may yield one and the same semantic interpretation. In **L***, each interpretation of a sequent corresponds to exactly one proof. In Section 5, we use **L*** to prove that an explicit elaboration of the suggestions in Moortgat's 'Unambiguous Proof Representations for the Lambek Calculus' (1990) leads to representations of **L** proofs of sequents with atomic goal categories which 'summarize' **L*** derivations and are, hence, indeed 'unambiguous', i.e., devoid of spurious ambiguity. A comparison of **L*** with the proof nets proposed in Roorda (1991), on the other hand, shows that every **L*** proof corresponds to a non-singleton *set* of proof nets. Finally, with the help of **L*** we are able to show in Section 6 that syntactic recognition within **L** can be mimicked using one basic category only, thus extending a result of Ponse (1988) for **LP**, the Lambek calculus with permutation, to **L** proper.

1 Introduction

We will employ the following definitions of the notions of category and sequent:

(1) Let ATOM be some finite set of atomic categories (none of which is of the form (a/b) or $(b\backslash a)$). Then CAT, the set of *categories based on* ATOM, is the smallest set such that (i) ATOM $\subseteq$ CAT, and (ii) if $a \in$ CAT and $b \in$ CAT, then $(a/b) \in$ CAT and $(b\backslash a) \in$ CAT.

(2) A *sequent* is an expression $T \vdash c$, where T is a finite non-empty sequence of categories and $c \in$ CAT. (So, $T = c_1, \ldots, c_n$, where $n > 0$ and for all i, $1 \leq i \leq n$: $c_i \in$ CAT.)

Outermost brackets of categories will be omitted. The sequence of categories $c_1, \ldots, c_n$ will be called the *left-hand side* of a sequent $c_1, \ldots, c_n \vdash c$, and the single category c its *right-hand side* or *goal*.

The product-free Lambek-Gentzen sequent calculus **L** consists of a set of axioms plus five inference rules: $/L$, $\backslash L$, $/R$, $\backslash R$ and *Cut*. The axioms and rules are listed in (3) and (4), respectively. In (4), a, b and c denote arbitrary categories and T, U and V arbitrary finite sequences of categories, of which T is non-empty.

(3) AXIOM, the set of *axioms* of **L**, is the set $\{c \vdash c \mid c \in \text{CAT}\}$.

(4) $\dfrac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c}\,[/L]$ $\qquad$ $\dfrac{T \vdash b \quad U, a, V \vdash c}{U, T, b\backslash a, V \vdash c}\,[\backslash L]$

$\dfrac{T, b \vdash a}{T \vdash a/b}\,[/R]$ $\qquad$ $\dfrac{b, T \vdash a}{T \vdash b\backslash a}\,[\backslash R]$

$\dfrac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$

Moortgat (1988) proposes a semantics for **L** in which every category occurrence c in a derivation is assigned some typed lambda term γ. The type of γ depends on c. For categories $c \in$ ATOM, some type assignment TYPE(c) is assumed, and the function TYPE is extended to non-atomic categories by the stipulation that TYPE(a/b) = TYPE($b\backslash a$) = (TYPE(b), TYPE(a)). In this semantics, which is given in (5) below, the axioms of **L** correspond to *identity*, the rules $/L$ and $\backslash L$ to *functional application*, the rules $/R$ and $\backslash R$ to *lambda abstraction*, and the *Cut* rule to *substitution*.[1] The respective assignment of terms $\gamma_1, \ldots, \gamma_n, \gamma$ to categories $c_1, \ldots, c_n, c$ in a derivable sequent $c_1, \ldots, c_n \vdash c$ is usually written as $c_1 : \gamma_1, \ldots, c_n : \gamma_n \vdash c : \gamma$.

(5)

SYNTAX:	SEMANTICS:
$c \vdash c$ [AXIOM]	$\gamma \vdash \gamma$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c}\,[/L]$	$\dfrac{T' \vdash \beta \quad U', \varphi(\beta), V' \vdash \gamma}{U', \varphi, T', V' \vdash \gamma}$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, T, b\backslash a, V \vdash c}\,[\backslash L]$	$\dfrac{T' \vdash \beta \quad U', \varphi(\beta), V' \vdash \gamma}{U', T', \varphi, V' \vdash \gamma}$
$\dfrac{T, b \vdash a}{T \vdash a/b}\,[/R]$	$\dfrac{T', v \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{b, T \vdash a}{T \vdash b\backslash a}\,[\backslash R]$	$\dfrac{v, T' \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$	$\dfrac{T' \vdash \alpha \quad U', \alpha, V' \vdash \gamma}{U', T', V' \vdash \gamma}$

[1] Viz., the substitution of T' (which yields α) for α in $U', \alpha, V' \vdash \gamma$.

In (5), only the terms of the 'active' categories are indicated separately. The expression v denotes a variable of type TYPE(b), while α, β, γ and φ denote terms of type TYPE(a), TYPE(b), TYPE(c) and (TYPE(b), TYPE(a)), respectively. The expressions T', U' and V' refer to the sequences of terms assigned to the sequences of 'inactive' categories T, U and V. Thus, the terms assigned to the categories in T, U and V are identical in premise and conclusion.

The interpretation which a proof with conclusion $c_1{:}\gamma_1, \ldots, c_n{:}\gamma_n \vdash c{:}\gamma$ assigns to a sequence of lexical expressions $e_1, \ldots, e_n$ of categories $c_1, \ldots, c_n$ with interpretations $e'_1, \ldots, e'_n$ is obtained by performing the simultaneous substitution $[\gamma_1{:=}e'_1, \ldots, \gamma_n{:=}e'_n]$ in the term γ assigned to the goal of the conclusion sequent: $\gamma[\gamma_1{:=}e'_1, \ldots, \gamma_n{:=}e'_n]$.

For example: let the lexical expressions *someone*, *bores* and *everyone* be assigned the categories $(s/n)\backslash s$, $(n\backslash s)/n$ and $(s/n)\backslash s$, as well as the interpretations $\lambda P.\exists z\, P(z)$, BORE and $\lambda P.\forall z\, P(z)$ of type $((e,t),t)$, $(e,(e,t))$ and $((e,t),t)$, respectively. Then the following derivation shows that *Someone bores everyone* is of category s:

(6)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \qquad \dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s \vdash s}[\backslash L]}{n, (n\backslash s)/n, n \vdash s}[/L]}{n, (n\backslash s)/n \vdash s/n}[/R] \qquad s \vdash s}{n, (n\backslash s)/n, (s/n)\backslash s \vdash s}[\backslash L]}{(n\backslash s)/n, (s/n)\backslash s \vdash n\backslash s}[\backslash R] \qquad s \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s}[/L]$$

An interpretation of derivation (6) is given in (7), where u, v, w, x, and y represent variables of type e, e, $((e,t),t)$, $(e,(e,t))$ and $((e,t),t)$, respectively:

(7)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{v \vdash v \qquad \dfrac{u \vdash u \quad x(v)(u) \vdash x(v)(u)}{u, x(v) \vdash x(v)(u)}}{u, x, v \vdash x(v)(u)}}{u, x \vdash \lambda v.x(v)(u)} \qquad y(\lambda v.x(v)(u)) \vdash y(\lambda v.x(v)(u))}{u, x, y \vdash y(\lambda v.x(v)(u))}}{x, y \vdash \lambda u.y(\lambda v.x(v)(u))} \qquad w(\lambda u.y(\lambda v.x(v)(u))) \vdash w(\lambda u.y(\lambda v.x(v)(u)))}{w, x, y \vdash w(\lambda u.y(\lambda v.x(v)(u)))}$$

Performing the simultaneous substitution (8) in $w(\lambda u.y(\lambda v.x(v)(u)))$, the term assigned to the conclusion goal in (7), supplies the sentence *Someone bores everyone* with an interpretation equivalent to $\exists u \forall v \, \text{BORE}(v)(u)$.

(8) $[w := \lambda P.\exists z \, P(z), x := \text{BORE}, y := \lambda P.\forall z \, P(z)]$

Note, first, that semantics (5) needs to be supplemented with at least the following condition, which guarantees that the abstractions introduced by $\backslash R$ and $/R$ bind only one variable occurrence:

(9) If $T : T' \vdash c : \lambda v.\gamma$ is the conclusion of a $\backslash R$ or $/R$ inference, then v does not occur freely in T'.

For instance, if – violating this condition but in keeping with (5) – all occurrences of v in (7) are replaced by u, then the conclusion goal of (6) is assigned the term $w(\lambda u.y(\lambda u.x(u)(u)))$, which results in the incorrect interpretation $\exists u \forall u \, \text{BORE}(u)(u)$, 'Everyone bores himself', for *Someone bores everyone*.

And, second, note that simultaneous substitution is not always well-defined for semantics (5), which allows complex terms and different occurrences of the same term in the left-hand side of sequents. For (*i*) it is unclear what is meant by a substitution of a term for a complex term (this requires at least some extension of the notion of 'free occurrence of' to terms other than variables), and (*ii*) simultaneous substitution does not make sense when the same term is to be substituted twice. The problem reappears at deeper levels, e.g., when a left-hand side category c_i is assigned a term γ_i which also occurs as a subterm of a term γ_j assigned to another left-hand side category c_j. Moreover, this substitution problem causes a complication in the proof of 'semantic *Cut* elimination'.[2]

[2]Consider Moortgat's treatment (1990, p. 151) of the crucial case **5** of Lambek's *Cut* elimination algorithm (cf. section 2 below), in which:

$$\dfrac{\dfrac{[\text{A}]_{T,\, d:v \vdash b:\beta}}{T \vdash b/d : \lambda v.\beta}\,[/R] \qquad \dfrac{V_1 \vdash d:\delta \quad [\text{B}]_{U,\, b:[\lambda v.\beta](\delta),\, V_2 \vdash c:\gamma}}{U, b/d:\lambda v.\beta, V_1, V_2 \vdash c:\gamma}\,[/L]}{U, T, V_1, V_2 \vdash c:\gamma}\,[Cut]$$

is replaced by:

$$\dfrac{V_1 \vdash d:\delta \qquad \dfrac{[\text{A}']_{T,\, d:\delta \vdash b:\beta[v:=\delta]} \quad [\text{B}']_{U,\, b:\beta[v:=\delta],\, V_2 \vdash c:\gamma[[\lambda v.\beta](\delta):=\beta[v:=\delta]]}}{U, T, d:\delta, V_2 \vdash c:\gamma[[\lambda v.\beta](\delta):=\beta[v:=\delta]]}\,[Cut]}{U, T, V_1, V_2 \vdash c:\gamma[[\lambda v.\beta](\delta):=\beta[v:=\delta]]}\,[Cut]$$

(The terms assigned to U, T, V_1 and V_2 have been omitted.) In order to be able to make the transition from [A] and [B] to [A'] and [B'], respectively, Moortgat needs something

These considerations suggest that it is more natural to have a semantics which always yields proofs of sequents $c_1 : \gamma_1, \ldots, c_n : \gamma_n \vdash c : \gamma$ where the terms $\gamma_1, \ldots, \gamma_n$ constitute a sequence of different variables. Therefore, the alternative semantics (10) which will be used in the following sections is based on the same category-to-type assignment as semantics (5), but differs from (5) as regards the terms it assigns to the categories in a derivable sequent $c_1, \ldots, c_n \vdash c$. For the left-hand side categories c_i $(1 \leq i \leq n)$, this interpretation is some variable v_i of type $\text{TYPE}(c_i)$, whereas for the right-hand side category c, it is a (possibly complex) term γ of type $\text{TYPE}(c)$: $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma$.

In (10), u, v, x and w represent variables of type $\text{TYPE}(a)$, $\text{TYPE}(b)$, $\text{TYPE}(c)$ and $(\text{TYPE}(b), \text{TYPE}(a))$; α, β and γ denote terms of type $\text{TYPE}(a)$, $\text{TYPE}(b)$ and $\text{TYPE}(c)$; and T', U' and V' refer to the sequences of variables assigned to the sequences of categories T, U and V. Again, only the terms of the active categories are indicated separately, and the terms of other categories are identical in premise and conclusion. The expression $\gamma[u := \alpha]$

---

like: (\$) If $\pi[U : U', a : \alpha, V : V' \vdash_{(5)} c : \gamma]$, then for all terms α' of type $\text{TYPE}(a)$: $\pi[U : U', a : \alpha', V : V' \vdash_{(5)} c : \gamma[\alpha := \alpha']]$.

(We let $\pi[c_1 : \gamma_1, \ldots, c_n : \gamma_n \vdash_{(5)} c : \gamma]$ represent that semantics (5) assigns the respective terms $\gamma_1, \ldots \gamma_n, \gamma$ to the categories $c_1, \ldots, c_n, c$ in the conclusion sequent $c_1, \ldots, c_n \vdash c$ of π. Note that $[\alpha := \alpha']$ is more than mere substitution for free variable occurrences; it also involves replacement of compound terms – cf. the transition from [B] to [B'].) But such a lemma is not available, as can be seen from the following derivations – where α, α', β and γ are terms of type $\text{TYPE}(a)$, $\text{TYPE}(a)$, $\text{TYPE}((a\backslash b)/a)$ and $(\text{TYPE}(a), \text{TYPE}(a))$, respectively:

$$(i) \qquad \dfrac{a : \alpha \vdash a : \alpha \qquad \dfrac{a : \gamma(\alpha) \vdash a : \gamma(\alpha) \quad b : \beta(\alpha)(\gamma(\alpha)) \vdash b : \beta(\alpha)(\gamma(\alpha))}{b/a : \beta(\alpha), a : \gamma(\alpha) \vdash b : \beta(\alpha)(\gamma(\alpha))}\,[/L]}{a : \alpha, a\backslash(b/a) : \beta, a : \gamma(\alpha) \vdash b : \beta(\alpha)(\gamma(\alpha))}\,[\backslash L]$$

$$(ii) \qquad \dfrac{a : \alpha' \vdash a : \alpha' \qquad \dfrac{a : \gamma(\alpha) \vdash a : \gamma(\alpha) \quad b : \beta(\alpha')(\gamma(\alpha)) \vdash b : \beta(\alpha')(\gamma(\alpha))}{b/a : \beta(\alpha'), a : \gamma(\alpha) \vdash b : \beta(\alpha')(\gamma(\alpha))}\,[/L]}{a : \alpha', a\backslash(b/a) : \beta, a : \gamma(\alpha) \vdash b : \beta(\alpha')(\gamma(\alpha))}\,[\backslash L]$$

Since no variable-binding operator is present, it seems plausible to assume that both occurrences of α in the term assigned to the conclusion goal of derivation (i) are free and that, hence, $[\beta(\alpha)(\gamma(\alpha))][\alpha := \alpha']$ is the term $\beta(\alpha')(\gamma(\alpha'))$. But, refuting (\$), the term assigned to the conclusion goal of (ii) is $\beta(\alpha')(\gamma(\alpha))$.

If, however, condition (9) is met, then in every sequent $T : T' \vdash c : \gamma$ the sequence T' of left-hand side terms has the same free variables as γ, and the following can be established (where v is a variable and α is a term, both of type $\text{TYPE}(a)$):

(#) If $\pi[T : T' \vdash_{(5)} c : \gamma]$, then $\pi[T : T'[v := \alpha] \vdash_{(5)} c : \gamma[v := \alpha]]$.

denotes the result of substituting term α for all free occurrences of u in γ.

(10)

SYNTAX:	SEMANTICS:
$c \vdash c$ [AXIOM]	$x \vdash x$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c}\,[/L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', w, T', V' \vdash \gamma[u{:=}w(\beta)]}$
$\dfrac{T \vdash b \quad U, a, V \vdash c}{U, T, b\backslash a, V \vdash c}\,[\backslash L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', T', w, V' \vdash \gamma[u{:=}w(\beta)]}$
$\dfrac{T, b \vdash a}{T \vdash a/b}\,[/R]$	$\dfrac{T', v \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{b, T \vdash a}{T \vdash b\backslash a}\,[\backslash R]$	$\dfrac{v, T' \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut]$	$\dfrac{T' \vdash \alpha \quad U', u, V' \vdash \gamma}{U', T', V' \vdash \gamma[u{:=}\alpha]}$

In the context of a proof, we will assume that all variables x and w assigned to axiom instances or introduced in the conclusions of $/L$ or $\backslash L$ inferences are different. Observe that, consequently, (*i*) the variables $v_1, \ldots v_n$ assigned to the left-hand side of a sequent $c_1, \ldots, c_n \vdash c$ are all different; (*ii*) these variables $v_1, \ldots v_n$ make up the free variables of the term γ assigned to the goal c; and (*iii*) each variable $v_1, \ldots v_n$ occurs exactly once in γ.

Note that semantics (10) is more 'compositional' than (5), in that the complexity of the term γ assigned to the goal of a sequent $c_1, \ldots, c_n \vdash c$ increases with the length of the proof of that sequent. This property makes semantics (10) more perspicuous, which facilitates proving things about it. In semantics (5), on the other hand, we see that the rule $/L$ assigns a term φ to its conclusion category a/b which is less complex than the term $\varphi(\beta)$ assigned to the category a in its right-hand side premise. (The same holds for $\backslash L$.) Moreover, semantics (10) is easier to use in practice, since it entails considerably less duplication of semantic information than (5), witness its

interpretation of derivation (6):

(11)
$$\dfrac{\dfrac{v \vdash v \quad \dfrac{u \vdash u \quad z \vdash z}{u, z' \vdash z'(u)}}{u, x, v \vdash x(v)(u)}}{\dfrac{u, x \vdash \lambda v.x(v)(u) \qquad z'' \vdash z''}{\dfrac{u, x, y \vdash y(\lambda v.x(v)(u))}{\dfrac{x, y \vdash \lambda u.y(\lambda v.x(v)(u)) \qquad z''' \vdash z'''}{w, x, y \vdash w(\lambda u.y(\lambda v.x(v)(u)))}}}}$$

Still, to the extent that semantics (5) is well-behaved, it is essentially the same as semantics (10), on account of the following relationship. Suppose that the assignment of terms by (5) is restricted by condition (9), and, for $i \in \{5, 10\}$, let $\pi[c_1 : \gamma_1, \ldots, c_n : \gamma_n \vdash_{(i)} c : \gamma]$ represent that semantics (i) assigns the respective terms $\gamma_1, \ldots, \gamma_n, \gamma$ to the categories $c_1, \ldots, c_n, c$ in the conclusion sequent $c_1, \ldots, c_n \vdash c$ of proof π. Then the following holds:

(12) $\pi[c_1 : v_1, \ldots, c_n : v_n \vdash_{(10)} c : \gamma]$ if and only if
$\pi[c_1 : \gamma_1, \ldots, c_n : \gamma_n \vdash_{(5)} c : \gamma[v_1 := \gamma_1 \ldots, v_n := \gamma_n]]$.

From (12), which is proven by induction on the length of π (where fact (#) – see footnote 2 – is needed as a lemma for the cases $\backslash R$ and $/R$), it follows that if the terms $\gamma_1, \ldots, \gamma_n$ of the left-hand side categories $c_1, \ldots, c_n$ in the conclusion sequent $c_1, \ldots, c_n \vdash c$ of π constitute a sequence $v_1, \ldots, v_n$ of different variables, then semantics (10) assigns a term to the goal category c which is syntactically identical to the term assigned by semantics (5).

2 Cut Elimination and Semantics

Van Benthem (1986) proved that the number of distinct readings for an **L**-derivable sequent is finite. However, in addition one would like to have an effective method for generating all readings. The availability of such a method could serve practical purposes by providing a solution to the 'parsing problem', which is:

> ...to find all readings for a given string. This is a necessary step if one wants to give a satisfactory parsing algorithm for **L**. It should be noticed, however, that it is not at all clear which members of the infinite set of derivations for a sentence S in **L**

> lead to different readings, and thus, finding all possible readings seems to be problematic to begin with, let alone finding them efficiently. (Bouma (1989))

A possible way of solving this problem is to prove that the result of applying Lambek's *Cut* elimination algorithm is a derivation which is semantically equivalent to the original derivation. Then for every derivation of an **L**-derivable sequent there is a *Cut*-free derivation with the same interpretation, and we can safely restrict our attention to the (finite number of) *Cut*-free derivations. Each of the inference rules $/L$, $\backslash L$, $/R$ and $\backslash R$ derives its conclusion from one or more premises with a strictly smaller number of occurrences of / and \. Hence establishing the derivability of the premise(s) is more simple than establishing the derivability of the conclusion, and it follows that every sequent has only finitely many *Cut*-free derivations.

Adapting the proof of Lambek (1958), we will show in this section that sequents in the proof of which the *Cut* inference rule has been used can also be derived without using *Cut*, while keeping their semantics the same. That is, *Cut* is a rule which does not yield new theorems, nor new interpretations. This is done in a constructive way: it is possible to transform every proof which makes use of *Cut* into a *Cut*-free proof which determines the same interpretation as the original one (see also Lambek (1990), Moortgat (1990)).

The two base cases of this transformation procedure (case **1** and **2** below) involve *Cut* inferences where one of the premises is an instance of the axiom scheme $c \vdash c$. Here the conclusion coincides with the other premise, and the *Cut* inference can be eliminated immediately. The recursive cases of the procedure (case **3**, **4** and **5** below) work by reduction of what is called the 'degree' of the *Cut* inference. Degree is defined as follows:

(13) (*i*) The degree $d(c)$ of a category c is defined inductively: $d(c) = 0$ for $c \in$ ATOM; $d(a/b) = d(b\backslash a) = d(a) + d(b) + 1$.

(*ii*) The degree $d(c_1, \ldots, c_n)$ of a finite sequence of categories $c_1, \ldots, c_n$ equals $d(c_1) + \ldots + d(c_n)$.

(*iii*) The degree $d(T \vdash c)$ of a sequent $T \vdash c$ equals $d(T) + d(c)$.

(*iv*) The degree $d(\alpha)$ of a *Cut* inference $\alpha =$

$$\frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}$$

equals $d(T) + d(U) + d(V) + d(a) + d(c)$.

Thus, the degree of a category, a sequence of categories and a sequent is equal to the number of slashes it contains.

Any *Cut* inference of which at least one premise has been proven without *Cut* can be turned into one or two new *Cut* inferences of strictly smaller degree. Since the minimal degree of an inference is zero, these proof transformations will ultimately converge on the base case.

We will now prove

1. that in any application of *Cut*,

$$\frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}\,[Cut],$$

   of which at least one premise has been proven without *Cut*,

   (a) either the conclusion is identical with one of the premises so that the application of *Cut* can be eliminated,

   (b) or the application of *Cut* can be replaced by one or two applications of *Cut* of smaller degree; and

2. that execution of this *Cut* elimination procedure always leads to semantically equivalent proofs.

There are five main cases:

1: $T \vdash a$ is an axiom; then $T = a$ and the conclusion coincides with the premise $U, a, V \vdash c$.

$$\frac{a \vdash a \quad U, a, V \vdash c}{U, a, V \vdash c}\,[Cut] \qquad \frac{u' \vdash u' \quad U', u, V' \vdash \gamma}{U', u', V' \vdash \gamma[u := u']}\,[Cut]$$

Cut can be eliminated, for note that the variable u' does not occur in γ. Hence we also have the interpretation $U', u', V' \vdash \gamma[u := u']$ for (the subproof of) the premise $U, a, V \vdash c$, due to the following fact (which is proven by a straightforward induction on the length of π):

> If $\pi[U : U', a : u, V : V' \vdash c : \gamma]$ and u' is a variable of type TYPE(a) not occurring in γ, then $\pi[U : U', a : u', V : V' \vdash c : \gamma[u := u']]$.

2: $U, a, V \vdash c$ is an axiom; then U and V are empty, $c = a$, and the conclusion coincides with the premise $T \vdash a$. *Cut* can be eliminated because the terms α and $u[u{:=}\alpha]$ assigned to the coinciding premise and conclusion coincide as well:

$$\frac{T \vdash a \quad a \vdash a}{T \vdash a}\,[Cut] \qquad\qquad \frac{T' \vdash \alpha \quad u \vdash u}{T' \vdash u[u{:=}\alpha]}\,[Cut]$$

3: The last step in the proof of $T \vdash a$ uses a rule but does not introduce the main connective of a. Then $T \vdash a$ is derived by $/L$ (or $\backslash L$) from two sequents, one of which has the form $T' \vdash a$, with $d(T') < d(T)$. Hence we can reverse the order of the rules $/L$ (or $\backslash L$) and *Cut*, resulting in a new *Cut* inference which has smaller degree than the old one. We only consider $/L$. ($\backslash L$ is analogous.) Then $T = P, b/d, Q, R$, and we can replace:

$$\frac{\dfrac{Q \vdash d \quad P, b, R \vdash a}{P, b/d, Q, R \vdash a}\,[/L] \qquad U, a, V \vdash c}{U, P, b/d, Q, R, V \vdash c}\,[Cut]$$

by:

$$\frac{Q \vdash d \qquad \dfrac{P, b, R \vdash a \quad U, a, V \vdash c}{U, P, b, R, V \vdash c}\,[Cut]}{U, P, b/d, Q, R, V \vdash c}\,[/L]$$

Semantically, this amounts to replacing:

$$\frac{\dfrac{Q' \vdash \delta \qquad P', v, R' \vdash \alpha}{P', w, Q', R' \vdash \alpha[v{:=}w(\delta)]}\,[/L] \qquad U', u, V' \vdash \gamma}{U', P', w, Q', R', V \vdash \gamma[u{:=}\alpha[v{:=}w(\delta)]]}\,[Cut]$$

by:

$$\frac{Q' \vdash \delta \qquad \dfrac{P', v, R' \vdash \alpha \quad U', u, V' \vdash \gamma}{U', P', v, R', V' \vdash \gamma[u{:=}\alpha]}\,[Cut]}{U', P', w, Q', R', V' \vdash [\gamma[u{:=}\alpha]][v{:=}w(\delta)]}\,[/L]$$

The terms $\gamma[u{:=}\alpha[v{:=}w(\delta)]]$ and $[\gamma[u{:=}\alpha]][v{:=}w(\delta)]$ are identical because v does not occur in γ; hence this replacement of *Cut* preserves interpretation.

4: The last step in the proof of $U, a, V \vdash c$ uses one of the rules $/L$, $\backslash L$, $/R$, $\backslash R$, but does not introduce the main connective of a. Then $U, a, V \vdash c$ is inferred from one or two sequents, one of which has the form $U', a, V' \vdash c'$ with $d(U')+d(V')+d(c') < d(U)+d(V)+d(c)$ (since the inference introduces one occurrence of a connective). Again, we can reverse the order of the rules; the new *Cut* inference has smaller degree than the given one. We only consider $\backslash L$ and $\backslash R$. ($/L$ and $/R$ are analogous.)

4.1: $U, a, V \vdash c$ is obtained by $\backslash L$. Then $U, a, V = P, Q, d\backslash b, R$. Note that a and $d\backslash b$ are distinct category occurrences (for $\backslash L$ does not introduce the main connective of a). Therefore, there are three possibilities: a is part of P, Q or R (The last subcase, which is analogous to the first one, will not be treated.)

4.1.a: The category a is part of P, so $P = P_1, a, P_2$. Replace:

$$\dfrac{T \vdash a \quad \dfrac{Q \vdash d \quad P_1, a, P_2, b, R \vdash c}{P_1, a, P_2, Q, d\backslash b, R \vdash c}\,[\backslash L]}{P_1, T, P_2, Q, d\backslash b, R \vdash c}\,[Cut]$$

by:

$$\dfrac{Q \vdash d \quad \dfrac{T \vdash a \quad P_1, a, P_2, b, R \vdash c}{P_1, T, P_2, b, R \vdash c}\,[Cut]}{P_1, T, P_2, Q, d\backslash b, R \vdash c}\,[\backslash L]$$

On the semantic side, this entails the replacement of:

$$\dfrac{T' \vdash \alpha \quad \dfrac{Q' \vdash \delta \qquad P_1', u, P_2', v, R' \vdash \gamma}{P_1', u, P_2', Q', w, R' \vdash \gamma[v{:=}w(\delta)]}\,[\backslash L]}{P_1', T', P_2', Q', w, R' \vdash [\gamma[v{:=}w(\delta)]][u{:=}\alpha]}\,[Cut]$$

by:

$$\dfrac{Q' \vdash \delta \quad \dfrac{T' \vdash \alpha \quad P_1', u, P_2', v, R' \vdash \gamma}{P_1', T', P_2', v, R' \vdash \gamma[u{:=}\alpha]}\,[Cut]}{P_1', T', P_2', Q', w, R' \vdash [\gamma[u{:=}\alpha]][v{:=}w(\delta)]}\,[\backslash L]$$

The terms $[\gamma[v{:=}w(\delta)]][u{:=}\alpha]$ and $[\gamma[u{:=}\alpha]][v{:=}w(\delta)]$ are identical, because u and v are different variables, u does not occur in $w(\delta)$, and v does not occur in α.

4.1.b: The category a is part of Q, so $Q = Q_1, a, Q_2$. Replace:

$$\dfrac{T \vdash a \quad \dfrac{Q_1, a, Q_2 \vdash d \quad P, b, R \vdash c}{P, Q_1, a, Q_2, d\backslash b, R \vdash c}\,[\backslash L]}{P, Q_1, T, Q_2, d\backslash b, R \vdash c}\,[Cut]$$

by:

$$\dfrac{T \vdash a \quad \dfrac{Q_1, a, Q_2 \vdash d}{Q_1, T, Q_2 \vdash d}\,[Cut] \qquad P, b, R \vdash c}{P, Q_1, T, Q_2, d\backslash b, R \vdash c}\,[\backslash L]$$

This entails a semantic replacement of:

$$\dfrac{T' \vdash \alpha \quad \dfrac{Q'_1, u, Q'_2 \vdash \delta \qquad P', v, R' \vdash \gamma}{P', Q'_1, u, Q'_2, w, R' \vdash \gamma[v{:=}w(\delta)]}\,[\backslash L]}{P', Q'_1, T', Q'_2, w, R' \vdash [\gamma[v{:=}w(\delta)]][u{:=}\alpha]}\,[Cut]$$

by:

$$\dfrac{\dfrac{T' \vdash \alpha \quad Q'_1, u, Q'_2 \vdash \delta}{Q'_1, T', Q'_2 \vdash \delta[u{:=}\alpha]}\,[Cut] \qquad P', v, R' \vdash \gamma}{P', Q'_1, T', Q'_2, w, R' \vdash \gamma[v{:=}w(\delta[u{:=}\alpha])]}\,[\backslash L]$$

Again, the terms $[\gamma[v{:=}w(\delta)]][u{:=}\alpha]$ and $\gamma[v{:=}w(\delta[u{:=}\alpha])]$ are identical, for u does not occur in γ, and u and w are different variables.

4.2: $U, a, V \vdash c$ is obtained by $\backslash R$. Then $c = d\backslash b$, and we can replace:

$$\dfrac{T \vdash a \quad \dfrac{d, U, a, V \vdash b}{U, a, V \vdash d\backslash b}\,[\backslash R]}{U, T, V \vdash d\backslash b}\,[Cut]$$

by:

$$\dfrac{T \vdash a \quad \dfrac{d, U, a, V \vdash b}{d, U, T, V \vdash b}\,[Cut]}{U, T, V \vdash d\backslash b}\,[\backslash R]$$

This is reflected in the semantic replacement of:

$$\dfrac{T' \vdash \alpha \quad \dfrac{v, U', u, V' \vdash \beta}{U', u, V' \vdash \lambda v.\beta}\,[\backslash R]}{U', T', V' \vdash [\lambda v.\beta][u{:=}\alpha]}\,[Cut]$$

by:

$$\dfrac{\dfrac{T' \vdash \alpha \quad v, U', u, V' \vdash \beta}{v, U', T', V' \vdash \beta[u{:=}\alpha]}\,[Cut]}{U', T', V' \vdash \lambda v.[\beta[u{:=}\alpha]]}\,[\backslash R]$$

Since u and v are different variables, the term $[\lambda v.\beta][u{:=}\alpha]$ is identical to $\lambda v.[\beta[u{:=}\alpha]]$.

5: The last steps in the proofs of both $T \vdash a$ and $U, a, V \vdash c$ introduce the main connective, / or \, of a. So $a = b/d$ or $a = d\backslash b$. (Only the /-case is treated here.) We can replace:

$$\dfrac{\dfrac{T, d \vdash b}{T \vdash b/d}\,[/R] \quad \dfrac{V_1 \vdash d \quad U, b, V_2 \vdash c}{U, b/d, V_1, V_2 \vdash c}\,[/L]}{U, T, V_1, V_2 \vdash c}\,[Cut]$$

by:

$$\dfrac{V_1 \vdash d \quad \dfrac{T, d \vdash b \quad U, b, V_2 \vdash c}{U, T, d, V_2 \vdash c}\,[Cut]}{U, T, V_1, V_2 \vdash c}\,[Cut]$$

Semantically, this involves a replacement of:

$$\dfrac{\dfrac{T', v \vdash \beta}{T' \vdash \lambda v.\beta}\,[/R] \quad \dfrac{V_1' \vdash \delta \qquad U', u, V_2' \vdash \gamma}{U', w, V_1', V_2' \vdash \gamma[u := w(\delta)]}\,[/L]}{U', T', V_1', V_2' \vdash [\gamma[u := w(\delta)]][w := \lambda v.\beta]}\,[Cut]$$

by:

$$\dfrac{V_1' \vdash \delta \quad \dfrac{T', v \vdash \beta \quad U', u, V_2' \vdash \gamma}{U', T', v, V_2' \vdash \gamma[u := \beta]}\,[Cut]}{U', T', V_1', V_2' \vdash [\gamma[u := \beta]][v := \delta]}\,[Cut]$$

Because w does not occur in γ, the term $[\gamma[u := w(\delta)]][w := \lambda v.\beta]$ is identical to $\gamma[u := [w(\delta)][w := \lambda v.\beta]]$. The latter term is identical to $\gamma[u := [\lambda v.\beta](\delta)]$, because w does not occur in δ. Since the free variables V_1' of δ do not occur in β, they are free for v in β and we can apply β-conversion, with the result $[\gamma[u := \beta[v := \delta]]]$.[3] Since v does not occur in γ, this is the same term as $[\gamma[u := \beta]][v := \delta]$.

3 Further Normalization

We have shown that if $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma$ is derivable in **L**, then there is a *Cut*-free **L** derivation of $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma'$ such that γ and γ' are

[3] Thus, case **5** does not involve identity of terms, but equivalence under *β-conversion* (also called λ-conversion): $[\lambda v.\gamma'](\gamma'') = \gamma'[v := \gamma'']$ if γ'' is free for v in γ'.

It takes an induction on the length of the derivation to see that terms γ assigned to conclusion goals in *Cut*-free proofs are always in *β-normal form*, i.e., they do not have subterms of the form $[\lambda v.\gamma'](\gamma'')$.

equivalent. Consequently, we only have to consider the *Cut*-free derivations of a sequent to obtain all its semantic interpretations. But in spite of the fact that a given sequent has only finitely many *Cut*-free derivations, *Cut*-less **L** still suffers from what has been called the 'spurious ambiguity problem' (König (1989)), viz., the problem that different proofs of a given sequent may yield one and the same semantic interpretation. When the calculus is actually used for parsing and interpreting expressions, this may entail superfluous work: interpretations will be computed n times instead of once. For example, the sequent $c/c, c/c, c \vdash c$ has two semantically equivalent *Cut*-free proofs, viz., (14) and (15), in which the symbol $^\sharp$ is used to keep track of the active category:

(14)
$$\dfrac{\dfrac{\dfrac{c \vdash c \quad c \vdash c}{c/c^\sharp, c \vdash c}[/L] \qquad c \vdash c}{c/c^\sharp, c/c, c \vdash c}[/L] \qquad \dfrac{\dfrac{x \vdash x \quad u \vdash u}{y, x \vdash y(x)}[/L] \qquad v \vdash v}{z, y, x \vdash z(y(x))}[/L]$$

(15)
$$\dfrac{c \vdash c \quad \dfrac{c \vdash c \quad c \vdash c}{c/c^\sharp, c \vdash c}[/L]}{c/c, c/c^\sharp, c \vdash c}[/L] \qquad \dfrac{x \vdash x \quad \dfrac{u \vdash u \quad v \vdash v}{z, u \vdash z(u)}[/L]}{z, y, x \vdash z(y(x))}[/L]$$

Of course, not all ambiguity in **L** is spurious. Some sequents are really ambiguous. The sequent $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$, for example, has six *Cut*-free derivations, which are *not* all equivalent. These derivations are listed below. Each derivation is followed by the interpretation it assigns to its conclusion sequent $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$:

(16)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad \dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^\sharp \vdash s}[\backslash L]}{n, (n\backslash s)/n^\sharp, n \vdash s}[/L]}{n, (n\backslash s)/n \vdash s/n^\sharp}[/R] \qquad s \vdash s}{n, (n\backslash s)/n, (s/n)\backslash s^\sharp \vdash s}[\backslash L]}{(n\backslash s)/n, (s/n)\backslash s \vdash n\backslash s^\sharp}[\backslash R] \qquad s \vdash s}{s/(n\backslash s)^\sharp, (n\backslash s)/n, (s/n)\backslash s \vdash s}[/L]$$

$u, v, w \vdash u(\lambda u'.w(\lambda w'.v(w')(u')))$

(17)
$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad n\backslash s \vdash n\backslash s}{(n\backslash s)/n^{\sharp}, n \vdash n\backslash s}\,[/L] \qquad s \vdash s}{s/(n\backslash s)^{\sharp}, (n\backslash s)/n, n \vdash s}\,[/L]}{s/(n\backslash s), (n\backslash s)/n \vdash s/n^{\sharp}}\,[/R] \qquad s \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^{\sharp} \vdash s}\,[\backslash L]$$

$u, v, w \vdash w(\lambda w'.u(v(w')))$

(18)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^{\sharp} \vdash s}\,[\backslash L]}{n\backslash s \vdash n\backslash s^{\sharp}}\,[\backslash R]}{(n\backslash s)/n^{\sharp}, n \vdash n\backslash s} \quad s \vdash s}{s/(n\backslash s)^{\sharp}, (n\backslash s)/n, n \vdash s}\,[/L]}{s/(n\backslash s), (n\backslash s)/n \vdash s/n^{\sharp}}\,[/R] \qquad s \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^{\sharp} \vdash s}\,[\backslash L]$$

$u, v, w \vdash w(\lambda w'.u(\lambda u'.v(w')(u')))$

(19)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^{\sharp} \vdash s}\,[\backslash L]}{n \vdash n} \quad s \vdash s}{n, (n\backslash s)/n^{\sharp}, n \vdash s}\,[/L]}{(n\backslash s)/n, n \vdash n\backslash s^{\sharp}}\,[\backslash R]}{s/(n\backslash s)^{\sharp}, (n\backslash s)/n, n \vdash s}\,[/L]}{s/(n\backslash s), (n\backslash s)/n \vdash s/n^{\sharp}}\,[/R] \qquad s \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^{\sharp} \vdash s}\,[\backslash L]$$

$u, v, w \vdash w(\lambda w'.u(\lambda u'.v(w')(u')))$

(20)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n\backslash s \vdash n\backslash s \quad s \vdash s}{n \vdash n \quad s/(n\backslash s)^{\sharp}, n\backslash s \vdash s}\,[/L]}{s/(n\backslash s), (n\backslash s)/n^{\sharp}, n \vdash s}\,[/L]}{s/(n\backslash s), (n\backslash s)/n \vdash s/n^{\sharp}}\,[/R] \qquad s \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^{\sharp} \vdash s}\,[\backslash L]$$

$u, v, w \vdash w(\lambda w'.u(v(w')))$

(21)
$$\dfrac{\dfrac{\dfrac{\dfrac{n \vdash n \quad s \vdash s}{n, n\backslash s^{\sharp} \vdash s}[\backslash L]}{n\backslash s \vdash n\backslash s^{\sharp}}[\backslash R] \qquad s \vdash s}{n \vdash n \qquad s/(n\backslash s)^{\sharp}, n\backslash s \vdash s}[/L]}{\dfrac{s/(n\backslash s), (n\backslash s)/n^{\sharp}, n \vdash s}{\dfrac{s/(n\backslash s), (n\backslash s)/n \vdash s/n^{\sharp}}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^{\sharp} \vdash s}[\backslash L]}[/R]}[/L]$$

$$u, v, w \vdash w(\lambda w'.u(\lambda u'.v(w')(u')))$$

The terms assigned to the conclusions of the derivations (18), (19) and (21) are identical. Moreover, because the subterm $\lambda u'.v(w')(u')$ of the term $w(\lambda w'.u(\lambda u'.v(w')(u')))$ assigned to the conclusion goal of these derivations is equivalent to $v(w')$ by η-conversion,[4] the interpretation of these conclusions is equivalent to $u, v, w \vdash w(\lambda w'.u(v(w')))$, the interpretation assigned to the conclusions of (17) and (20). So, five of the six *Cut*-free derivations of the sequent $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash \vdash s$ are semantically equivalent.

However, the interpretation $u(\lambda u'.w(\lambda w'.v(w')(u')))$ of the conclusion of derivation (16) is *not* equivalent to the interpretation of the other conclusions. This can be seen as follows. If the expressions *someone*, *bores* and *everyone* are assigned categories and translations as in (22), then performing the simultaneous substitution (23) in the term of the conclusion goal in (16) supplies the sentence *Someone bores everyone* with the wide scope-subject interpretation $\exists x \forall y \, \text{BORE}(y)(x)$, whereas performing this substitution in the terms of the conclusion goals in (17) through (21) yields the wide scope-object interpretation $\forall y \exists x \, \text{BORE}(y)(x)$.

(22)

EXPRESSION	CATEGORY	INTERPRETATION
someone	$s/(n\backslash s)$	$\lambda P.\exists x \, P(x)$
bores	$(n\backslash s)/n$	BORE
everyone	$(s/n)\backslash s$	$\lambda P.\forall y \, P(y)$

[4] By *η-conversion*, the following equivalence is meant: $\lambda v.\gamma'(v) = \gamma'$, provided that v does not occur freely in γ'. A term γ is in *η-normal form* iff it does not have subterms of the form $\lambda v.\gamma'(v)$. Equivalence in typed λ-calculus amounts to identity of $\beta\eta$-normal forms in the following sense: for every term γ there is a unique term $\gamma^{\beta\eta}$ which is in $\beta\eta$-normal form (i.e., it is both in β-normal form and in η-normal form) and which can be obtained from γ by performing a finite number of β-conversions and η-conversions (where renaming of bound variables – i.e., the replacement of subterms $\lambda v.\alpha$ by $\lambda v'.[\alpha[v:=v']]$ if v' is free for v in α – is allowed), and two terms γ and δ are equivalent iff their $\beta\eta$-nfs $\gamma^{\beta\eta}$ and $\delta^{\beta\eta}$ are syntactically identical (again modulo renaming bound variables).

(23) $[u{:=}\lambda P.\exists x\, P(x), v{:=}\textsc{bore}, w{:=}\lambda P.\forall y\, P(y)]$

This example gives a nice illustration of what spurious ambiguity comes down to: the wide scope-subject interpretation is derived only once, but the wide-scope-object interpretation is associated with five derivations. Clearly, if we want to put the calculus **L** to use in parsing and interpreting expressions, this is going to be a problem: the latter interpretation will be computed five times instead of once.

König (1989) is the first attempt to solve the problem, but it does not succeed completely. For instance, König's system allows the two derivations (14) and (15) for the unambiguous sequent $c/c, c/c, c \vdash c$. In the sequel we will show how this problem can be solved for **L** by further restricting the *Cut*-free calculus. The resulting system **L***, which is equivalent to the system which was indepently proposed by Hepple (1990), is a solution to the spurious ambiguity problem: it provides exactly one proof per interpretation.

The 'non-spuriously ambiguous' calculus **L*** which will be presented below is based on the following three observations:

A: First, each non-atomic axiom instance $a/b \vdash a/b$ and $b\backslash a \vdash b\backslash a$ can be decomposed into a proof with two less complex axiom premises, $a \vdash a$ and $b \vdash b$. For instance, axioms $b\backslash a \vdash b\backslash a$ with interpretation $w \vdash w$ can be replaced by:

$$\dfrac{\dfrac{b \vdash b \quad a \vdash a}{b, b\backslash a \vdash a}\,[\backslash L]}{b\backslash a \vdash b\backslash a}\,[\backslash R] \qquad\qquad \dfrac{\dfrac{u \vdash u \quad v \vdash v}{u, w \vdash w(u)}\,[\backslash L]}{w \vdash \lambda u.w(u)}\,[\backslash R]$$

Due to η-conversion, the term $\lambda u.w(u)$ is equivalent to w. (Axioms $a/b \vdash a/b$ can be treated analogously.)

B: Second, if a $\backslash R$ or $/R$ inference yields the right-hand side premise of a $/L$ or $\backslash L$ inference, we can always reverse the order of the rules, whereas the semantics remains the same. We only consider the $\langle /R, /L\rangle$-case, where the rules involved are $/R$ and $/L$, respectively. (The cases $\langle /R, \backslash L\rangle$, $\langle \backslash R, /L\rangle$ and $\langle \backslash R, \backslash L\rangle$ are analogous.) We can replace:

(24)
$$\dfrac{T \vdash d \quad \dfrac{U, c, V, b \vdash a}{U, c, V \vdash a/b}\,[/R]}{U, c/d, T, V \vdash a/b}\,[/L]$$

by:

$$\dfrac{\dfrac{T \vdash d \quad U, c, V, b \vdash a}{U, c/d, T, V, b \vdash a}\,[/L]}{U, c/d, T, V \vdash a/b}\,[/R]$$

Semantically, this entails a replacement of:

$$\dfrac{T' \vdash \delta \quad \dfrac{U', z, V', y \vdash \alpha}{U', z, V' \vdash \lambda y.\alpha}\,[/R]}{U', v, T', V' \vdash [\lambda y.\alpha][z{:=}v(\delta)]}\,[/L]$$

by:

$$\dfrac{\dfrac{T' \vdash \delta \qquad U', z, V', y \vdash \alpha}{U', v, T', V', y \vdash \alpha[z{:=}v(\delta)]}\,[/L]}{U', v, T', V' \vdash \lambda y.[\alpha[z{:=}v(\delta)]]}\,[/R]$$

The terms $[\lambda y.\alpha][z{:=}v(\delta)]$ and $\lambda y.[\alpha[z{:=}v(\delta)]]$ are identical, since z and y are different variables.

C: Third, whenever a $\backslash L$ or $/L$ inference yields the right-hand side premise of another $\backslash L$ or $/L$ inference, and both inferences have *different* active categories,[5] we can reverse the order of the inferences and shift the latter inference to the left-hand side premise (cf. (25)) or to the right-hand side premise (cf. (26)) of the former one. We only treat the $\langle /L, \backslash L\rangle$-case. (The cases $\langle /L, /L\rangle$, $\langle \backslash L, /L\rangle$ and $\langle \backslash L, \backslash L\rangle$ are analogous.)

Suppose that the conclusion $X, W, d\backslash c, Z \vdash e$ has been derived by applying $\backslash L$ to left-hand side premise $W \vdash d$ and right-hand side premise $X, c, Z \vdash e$, and that the right-hand side premise $X, c, Z \vdash e$ is the result of

[5] The notions 'same' and 'different' active category are defined as follows: in a structure

$$\dfrac{T_2 \vdash b_2 \qquad \dfrac{T_1 \vdash b_1 \quad U, a, V \vdash c}{T_3 \vdash c}\,[\$L]}{T_4 \vdash c}\,[\#L]$$

(where \$ and # $\in \{\backslash, /\}$), the inferences $\$L$ and $\#L$ have the *same* active category iff

(*i*) $\$ = \backslash$, $\# = \backslash$, $T_3 = U, T_1, b_1\backslash a, V$ and $T_4 = U, T_1, T_2, b_2\backslash(b_1\backslash a), V$; or
(*ii*) $\$ = \backslash$, $\# = /$, $T_3 = U, T_1, b_1\backslash a, V$ and $T_4 = U, T_1, (b_1\backslash a)/b_2, T_2, V$; or
(*iii*) $\$ = /$, $\# = \backslash$, $T_3 = U, a/b_1, T_1, V$ and $T_4 = U, T_2, b_2\backslash(a/b_1), T_1, V$; or
(*iv*) $\$ = /$, $\# = /$, $T_3 = U, a/b_1, T_1, V$ and $T_4 = U, (a/b_1)/b_2, T_2, T_1, V$;

otherwise, the inferences $\$L$ and $\#L$ have *different* active categories.

applying $/L$ with an active category a/b different from c – that is: $X, c, Z = P, a/b, Q, R$, and the category a/b occurs in X or Z:

$$\dfrac{W \vdash d \quad \dfrac{Q \vdash b \quad P, a, R \vdash e}{X, c, Z \vdash e}\,[/L]}{X, W, d\backslash c, Z \vdash e}\,[\backslash L]$$

Then there are three possibilities: c occurs in Q (see (25)), or c does not occur in Q, and then c occurs in P (see (26)) or in R (analogous to (26)). In all cases we can reverse the order of the rules while keeping the same semantics. If c occurs in Q, then $Q = Q_1, c, Q_2$, and we can replace:

(25)
$$\dfrac{W \vdash d \quad \dfrac{Q_1, c, Q_2 \vdash b \quad P, a, R \vdash e}{P, a/b, Q_1, c, Q_2, R \vdash e}\,[/L]}{P, a/b, Q_1, W, d\backslash c, Q_2, R \vdash e}\,[\backslash L]$$

by:

$$\dfrac{\dfrac{W \vdash d \quad Q_1, c, Q_2 \vdash b}{Q_1, W, d\backslash c, Q_2 \vdash b}\,[\backslash L] \quad P, a, R \vdash e}{P, a/b, Q_1, W, d\backslash c, Q_2, R \vdash e}\,[/L]$$

Semantically, this entails the replacement of:

$$\dfrac{W' \vdash \delta \quad \dfrac{Q'_1, z, Q'_2 \vdash \beta \qquad P', x, R' \vdash \epsilon}{P', y, Q'_1, z, Q'_2, R' \vdash \epsilon[x{:=}y(\beta)]}\,[/L]}{P', y, Q'_1, W', v, Q'_2, R' \vdash [\epsilon[x{:=}y(\beta)]][z{:=}v(\delta)]}\,[\backslash L]$$

by:

$$\dfrac{\dfrac{W' \vdash \delta \qquad Q'_1, z, Q'_2 \vdash \beta}{Q'_1, W', v, Q'_2 \vdash \beta[z{:=}v(\delta)]}\,[\backslash L] \quad P', x, R' \vdash \epsilon}{P', y, Q'_1, W', v, Q'_2, R' \vdash \epsilon[x{:=}y(\beta[z{:=}v(\delta)])]}\,[/L]$$

The terms $[\epsilon[x{:=}y(\beta)]][z{:=}v(\delta)]$ and $\epsilon[x{:=}y(\beta[z{:=}v(\delta)])]$ are identical, because z does not occur in ϵ and z and y are different variables.

If c occurs in P, then $P = P_1, c, P_2$ and we can replace:

(26)
$$\dfrac{W \vdash d \quad \dfrac{Q \vdash b \quad P_1, c, P_2, a, R \vdash e}{P_1, c, P_2, a/b, Q, R \vdash e}\,[/L]}{P_1, W, d\backslash c, P_2, a/b, Q, R \vdash e}\,[\backslash L]$$

by:

$$\dfrac{Q \vdash b \quad \dfrac{W \vdash d \quad P_1, c, P_2, a, R \vdash e}{P_1, W, d\backslash c, P_2, a, R \vdash e}\,[\backslash L]}{P_1, W, d\backslash c, P_2, a/b, Q, R \vdash e}\,[/L]$$

In the semantics, this amounts to the replacement of:

$$\dfrac{W' \vdash \delta \quad \dfrac{Q' \vdash \beta \qquad P'_1, z, P'_2, x, R' \vdash \epsilon}{P'_1, z, P'_2, y, Q', R' \vdash \epsilon[x{:=}y(\beta)]}\,[/L]}{U', P'_1, v, P'_2, y, Q', R' \vdash [\epsilon[x{:=}y(\beta)]][z{:=}v(\delta)]}\,[\backslash L]$$

by:

$$\dfrac{Q' \vdash \beta \quad \dfrac{W' \vdash \delta \qquad P'_1, z, P'_2, x, R' \vdash \epsilon}{P'_1, W', v, P'_2, x, R' \vdash \epsilon[z{:=}v(\delta)]}\,[\backslash L]}{P'_1, W', v, P'_2, y, Q', R' \vdash [\epsilon[z{:=}v(\delta)]][x{:=}y(\beta)]}\,[/L]$$

The terms $[\epsilon[x{:=}y(\beta)]][z{:=}v(\delta)]$ and $[\epsilon[z{:=}v(\delta)]][x{:=}y(\beta)]$ are identical, since x and z are different variables, x does not occur in $v(\delta)$, and z does not occur in $y(\beta)$.[6]

Observation **A** entails that for every proof of a sequent, there is a semantically equivalent proof π in which all axiom instances are atomic: $at \vdash at$. For note that each non-atomic axiom instance can be decomposed in the way sketched above, and that every decomposition strictly decreases the sum of the degrees of the axiom instances. Hence every proof can be transformed into such an equivalent proof π after finitely many decompositions.

[6]Note (*i*) that transformation (26) cannot arise when the right-hand side premise of the first inference is an axiom, because the left-hand side of $P_1, c, P_2, a, R \vdash e$ contains at least the two categories c and a; and (*ii*) that if transformation (26) is executed in a configuration such as the one indicated below, where on the left-hand side $@L$ and $\$L$ have the same active category and $\$L$ and $\#L$ have different active categories, then on the right-hand-side $@L$ and $\#L$ (as well as $\#L$ and $\$L$) have different active categories:

$$\dfrac{T_5 \vdash c \quad \dfrac{T_3 \vdash b \quad \dfrac{T_1 \vdash a \quad T_2 \vdash d}{T_4 \vdash d}\,[@L]}{T_6 \vdash d}\,[\$L]}{T_7 \vdash d}\,[\#L] \quad \Rightarrow_{(26)} \quad \dfrac{T_3 \vdash b \quad \dfrac{T_5 \vdash c \quad \dfrac{T_1 \vdash a \quad T_2 \vdash d}{T_4 \vdash d}\,[@L]}{T'_6 \vdash d}\,[\#L]}{T_7 \vdash d}\,[\$L]$$

Moreover, given such a π, we can use observations **B** and **C** for obtaining a further normalized proof π'' in which (*i*) no *major* (right-hand side) premise of a $\backslash L$ or $/L$ inference is the conclusion of a $\backslash R$ or $/R$ inference (this corresponds to **B**);[7] and (*ii*) the same left-hand side category remains active whenever one goes down from axioms via major premises of $\backslash L$ and $/L$ inferences (this corresponds to **C**).

In order to see this, we first define the notion 'maximal major path' (mmp) for *Cut*-free **L** proofs π: an *mmp* is a sequence $\langle \sigma_0, \ldots, \sigma_p \rangle$ ($p \in \mathbb{N}$) of sequent occurrences in π, such that (*i*) σ_0 is an axiom instance; (*ii*) for all i, $0 \leq i < p$, σ_{i+1} is the conclusion of an inference with sole ($\backslash R$ or $/R$) or major ($\backslash L$ or $/L$) premise σ_i; and (*iii*) σ_p is not the sole or major premise of an inference.[8]

It is not hard to see that every sequent occurrence in a proof π is on exactly one mmp, and that every proof π contains as many mmps as it contains axiom instances, i.e., at least one.

Now, consider a proof π with $n+1$ mmps, and focus on the main mmp $\mu = \langle \sigma_0, \ldots, \sigma_p \rangle$ which culminates in the final conclusion σ_p of π. Let q be the number of $\backslash L$ and $/L$ inferences in μ, and r the number of $\backslash R$ and $/R$ inferences in μ. So, $\sigma_p = \sigma_{q+r}$, some sequent $T_{q+r} \vdash c_{q+r}$, and the sequent σ_0 is an axiom $c_0 \vdash c_0$. Moreover, for $0 \leq i \leq q+r$: (*a*) each σ_{i+1} obtained by $\backslash R$ or $/R$ is concluded from a subproof π_i of π with conclusion σ_i, and (*b*) each σ_{i+1} obtained by $\backslash L$ or $/L$ is concluded from two subproofs of π: π'_j (where $1 \leq j \leq q$) and π_i (with conclusion σ_i):

$$(a) \quad \dfrac{\left[\begin{array}{c}\vdots\\ \sigma_i\end{array}\right]\pi_i}{\sigma_{i+1}} \qquad\qquad (b) \quad \dfrac{\pi'_j \qquad \left[\begin{array}{c}\vdots\\ \sigma_i\end{array}\right]\pi_i}{\sigma_{i+1}}$$

The left-hand side subproofs π'_1 through π'_q have a total number of n mmps, which is one less than the number of mmps in the whole proof π.

Next, let $k(\mu) \leq r \cdot q$ be the number of pairs of sequents σ_i and σ_j in μ such that (*i*) $1 \leq i < j \leq q+r$; (*ii*) σ_i is the conclusion of a $\backslash R$ or $/R$ inference; and (*iii*) σ_j is the conclusion of a $\backslash L$ or $/L$ inference.

Note that an application of transformation (24) to a $\backslash R$ or $/R$ inference which yields the major premise of a $\backslash L$ or $/L$ inference in an mmp ν always

[7] Consequently, every major premise of a $\backslash L$ or $/L$ inference must be an (atomic) axiom instance $at \vdash at$ or the conclusion of another $\backslash L$ or $/L$ inference. Since $\backslash L$ and $/L$ identify the goal categories of their major premise and conclusion, every $\backslash L$ and $/L$ inference in π'' necessarily derives a conclusion sequent with an atomic goal category: $T \vdash at$.

[8] Therefore, σ_p must be either the *minor* (left-hand side) premise of a $\backslash L$ or $/L$ inference, or the final conclusion of π.

leads to an mmp ν' which (*i*) has the same length, (*ii*) has the same σ_{q+r}, (*iii*) builds on the same left-hand side subproofs $\pi'_1,\ldots\pi'_q$, and (*iv*) assigns the same interpretation to σ_{q+r} as ν; but that $k(\nu') = k(\nu) - 1$:

$$\dfrac{\pi'_j \quad \dfrac{\pi_{i-1}}{\sigma_i}}{\sigma_{i+1}} \quad \Rightarrow_{(24)} \quad \dfrac{\dfrac{\pi'_j \quad \pi_{i-1}}{\sigma'_i}}{\sigma_{i+1}}$$

So, after $k(\mu)$ applications of transformation (24) we obtain a semantically equivalent proof π' of $T_{q+r} \vdash c_{q+r}$ such that in the main mmp μ' of π' no $\backslash R$ or $/R$ inference precedes a $\backslash L$ or $/L$ inference, i.e., $k(\mu') = 0$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi'_1 \quad c_0 \vdash c_0}{T'_1 \vdash c'_1}\,[\text{AXIOM}]}{\vdots}}{\dfrac{\pi'_q}{T'_q \vdash c'_q}}}{\vdots}}{\dfrac{T'_{(q+r)-1} \vdash c'_{(q+r)-1}}{T_{q+r} \vdash c_{q+r}}} \quad \begin{matrix} \backslash L \text{ and } /L \\ \\ \backslash R \text{ and } /R \end{matrix}$$

Concentrate on the sequence $\nu = \langle c_0 \vdash c_0, \ldots, T'_q \vdash c'_q\rangle$ which is the main mmp of the subproof of π' with conclusion $T'_q \vdash c'_q$, and consider the first inference $\$L$ in ν which has a different active category than the inference $\#L$ which yields the major premise of $\$L$ (where $\$$ and $\# \in \{\backslash, /\}$). We can apply one of the proof transformations (25) and (26). If (26) is applicable, then inference $\$L$ reappears one place higher up in the main mmp, now yielding the major premise of $\#L$, and again having a different active category than the inference which now yields its major premise (cf. footnote 6 (*ii*)):

$$\dfrac{\pi'_{j+1} \quad \dfrac{\pi'_j \quad \pi_{i-1}}{\sigma_i}[\#L]}{\sigma_{i+1}}[\$L] \quad \Rightarrow_{(26)} \quad \dfrac{\pi'_j \quad \dfrac{\pi'_{j+1} \quad \pi_{i-1}}{\sigma'_i}[\$L]}{\sigma_{i+1}}[\#L]$$

Observe that an application of transformation (26) to an mmp of length q always leads to an mmp with (*i*) the same length, (*ii*) the same σ_q, (*iii*) the same left-hand side subproofs $\pi'_1,\ldots\pi'_q$ (though appearing in a different order), and (*iv*) the same interpretation for σ_q. Since the new situation is structurally the same as the old one, transformation (26) can be applied again and again, until transformation (25) is applicable – for $\$L$ yielding

conclusion σ_i $(2 \leq i \leq q)$, this happens after at most i steps: when $\$L$ yields σ_2, transformation (25) must be applied (cf. footnote 6 (i)).

Transformation (25) *removes* inference $\$L$ from the mmp and *adds* it, together with its left-hand side subproof π'_{j+1}, to the left-hand side subproof π'_j of $\#L$, where the number of mmps in the newly created left-hand side subproof π''_j is equal to the sum of the number of mmps in π'_j and π'_{j+1}:[9]

$$\dfrac{\pi'_{j+1}\quad \dfrac{\pi'_j \qquad \pi_{i-1}}{\sigma_i}[\$L]}{\sigma_{i+1}}[\#L] \quad \Rightarrow_{(25)} \quad \pi''_j\left[\dfrac{\dfrac{\pi'_{j+1} \qquad \pi'_j}{\sigma'_i}[\$L] \quad \pi_{i-1}}{\sigma_{i+1}}[\#L]\right.$$

The resulting mmp ν' is one sequent shorter than ν, but its last sequent σ'_{q-1} equals σ_q of ν, and receives the same interpretation. The same procedure can be reapplied to the first inference $\$L'$ in ν' which has a different active category than the inference $\#L'$ which yields its major premise. If this is done s times, where $0 \leq s \leq q-1$ and s is the number of $\backslash L$ and $/L$ inferences having a different active category than the first inference in ν, we obtain a semantically equivalent proof π'' of $T_{q+r} \vdash c_{q+r}$ such that in μ'', the main mmp of π'', no application of $\backslash R$ or $/R$ precedes an application of $\backslash L$ or $/L$, and all applications of $\backslash L$ and $/L$ have the same active category:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi''_1 \quad c_0 \vdash c_0}{T'_1 \vdash c'_1}\,[\text{AXIOM}]}{\vdots}\,\pi''_{q-s}}{T'_q \vdash c'_q}}{\vdots}}{T'_{(q+r)-1} \vdash c'_{(q+r)-1}}}{T_{q+r} \vdash c_{q+r}}$$

$\backslash L$ and $/L$: same active category

$\backslash R$ and $/R$

The finite procedure which led from π to π'' can be repeated for the subproofs $\pi''_1, \ldots, \pi''_{q-s}$ of π''. Moreover, the task of normalizing the proofs π''_1 through π''_{q-s} involves less work than our initial task of normalizing π, because we started with a proof with $n+1$ mmps, whereas the total number of mmps in π''_1 through π''_{q-s} equals n.

The above considerations can be summarized in the form of the calculus **L***, which is given in (27) below. This calculus introduces the symbol '*'

[9] So that the new series of left-hand side subproofs $\pi'_1, \ldots, \pi'_{j-1}, \pi''_j, \pi'_{j+2}, \ldots, \pi'_q$ of ν' still has a total number of n mmps.

which controls the activity of categories in derivations. Unlike the symbol '♯', which was merely added to the **L** derivations (14) through (21) in order to enhance their readability, the symbol '*' is a substantial ingredient of **L*** derivations, since there is a special rule which deals with its behaviour:

(27)

SYNTAX:	SEMANTICS:
$at^* \vdash at$ [AXIOM]	$x \vdash x$
$\dfrac{T \vdash b^* \quad U, a^*, V \vdash at}{U, a/b^*, T, V \vdash at}\,[/L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', w, T', V' \vdash \gamma[u{:=}w(\beta)]}$
$\dfrac{T \vdash b^* \quad U, a^*, V \vdash at}{U, T, b\backslash a^*, V \vdash at}\,[\backslash L]$	$\dfrac{T' \vdash \beta \qquad U', u, V' \vdash \gamma}{U', T', w, V' \vdash \gamma[u{:=}w(\beta)]}$
$\dfrac{T, b \vdash a^*}{T \vdash a/b^*}\,[/R]$	$\dfrac{T', v \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{b, T \vdash a^*}{T \vdash b\backslash a^*}\,[\backslash R]$	$\dfrac{v, T' \vdash \alpha}{T' \vdash \lambda v.\alpha}$
$\dfrac{U, a^*, V \vdash at}{U, a, V \vdash at^*}$	$\dfrac{U', u, V' \vdash \gamma}{U', u, V' \vdash \gamma}\,[*]$

The calculus in (27) observes the same conventions as regards categories, types, terms and variables as (10) above, with the exceptions that *at* represents an arbitrary atomic category, that γ refers to a term of type TYPE(at), and that x denotes a variable of type TYPE(at).

Important properties of the calculus **L*** are the following. Let $\gamma' = \gamma$ denote equivalence of terms γ' and γ (we will use $\gamma' \equiv \gamma$ for syntactic identity of terms modulo renaming bound variables):

Theorem 1:
1. $T \vdash_{\mathbf{L}^*} c^*$ if and only if $T \vdash_{\mathbf{L}} c$.
2. If $T{:}T' \vdash_{\mathbf{L}^*} c^*{:}\gamma$, then $T{:}T' \vdash_{\mathbf{L}} c{:}\gamma$.
3. If $T{:}T' \vdash_{\mathbf{L}} c{:}\gamma$, there is a γ' such that $\gamma' = \gamma$ and $T{:}T' \vdash_{\mathbf{L}^*} c^*{:}\gamma'$.

Proof: We have seen that if $T{:}T' \vdash_{\mathbf{L}} c{:}\gamma$, then there is a *Cut*-free **L** proof π of $T : T' \vdash c : \gamma'$ such that $\gamma' = \gamma$ and π has the following properties: (*a*) all axiom instances in π are atomic; (*b*) no major premise of a $\backslash L$ or $/L$

inference in π is the conclusion of a $\backslash R$ or $/R$ inference; and (c) the same category remains active whenever one goes down from axioms via major premises of $\backslash L$ and $/L$ inferences in π.

But this is sufficient, for there is an **L*** proof π' of $T:T' \vdash c^*:\gamma'$ iff there is a *Cut*-free **L** proof π of $T:T' \vdash c:\gamma'$ with the properties (a) through (c):

Observe (a') that **L*** axioms $at^* \vdash at$ involve only atomic categories; (b') that the major premise of a $\backslash L$ or $/L$ inference in **L*** can only be an axiom or the conclusion of another $\backslash L$ or $/L$ inference (the asterisk must be on the left-hand side); and (c') that if a $\backslash L$ or $/L$ inference yields the major premise of another $\backslash L$ or $/L$ inference, then they have the same (asterisked) active left-hand side category. On account of (a') through (c'), every *Cut*-free **L** proof π of a sequent $T:T' \vdash c:\gamma'$ having the properties (a) through (c) can be turned into an **L*** proof π' of $T:T' \vdash c^*:\gamma'$ by (i) adding an asterisk to the left-hand side category of every axiom instance; (ii) adding an asterisk to the active left-hand side category in every conclusion sequent of a $\backslash L$, $/L$, $\backslash R$ and $/R$ inference; and (iii) replacing every sequent of the form $U:U', a^*:u, V:V' \vdash at:\gamma$ which is not the major premise of a $\backslash L$ or $/L$ inference by the following inference:

$$(28)\quad \frac{U:U', a^*:u, V:V' \vdash at:\gamma}{U:U', a:u, V:V' \vdash at^*:\gamma}\,[*]$$

And, conversely, every **L*** proof π' of $T:T' \vdash c^*:\gamma'$ can be turned into a *Cut*-free **L** proof π of $T:T' \vdash c:\gamma'$ with properties (a) through (c) by (i) replacing every inference of the form (28) by the sequent $U:U', a:u, V:V' \vdash at:\gamma$; and (ii) deleting all remaining asterisks.

4 Spurious Ambiguity

In the present section we will show that **L*** is a solution to the spurious ambiguity problem: if π_1 is an **L*** proof of $T:T' \vdash c:\gamma$ and π_2 is an **L*** proof of $T:T' \vdash c:\gamma'$ such that $\gamma = \gamma'$, then $\pi_1 = \pi_2$.[10] As $\gamma = \gamma'$ entails that $\lambda T'.\gamma = \lambda T'.\gamma'$, this claim is a corollary of the following:

[10] Above we merely required that in the context of an **L** (or **L***) proof all variables x and w assigned to an axiom instance or introduced in the conclusion of a $\backslash L$ or $/L$ inference be different. Accordingly, we will identify all derivations which differ only with respect to the particular variables they assign. Thus for sequents σ and τ, we let $\sigma = \tau$ iff $\sigma = c_1:v_1, \ldots, c_n:v_n \vdash c:\gamma$ and $\tau = c_1:v_1', \ldots, c_n:v_n' \vdash c:\gamma'$; and for proofs π and ρ, we let $\pi = \rho$ if and only if

- $\pi = \sigma$, $\rho = \tau$, where σ and τ are (axiom) sequents such that $\sigma = \tau$;

Theorem 2:
If π_1 is an **L*** proof of $T:T' \vdash c:\gamma'$ and π_2 is an **L*** proof of $T:T'' \vdash c:\gamma''$ such that $\lambda T'.\gamma' = \lambda T''.\gamma''$, then $\pi_1 = \pi_2$.

Proof: Suppose π_1 and π_2 are two different proofs of the same conclusion $T \vdash c$. Since axioms and non-axiom theorems in **L*** never have the same degree (cf. definition (13) (*iii*) above), no sequent can be both an axiom and a non-axiom theorem. This entails that there must be at least one non-axiom sequent $T^\circ \vdash c^\circ$ which is derived from different premises in π_1 and π_2. Inspection of **L*** shows that the goal c° of this sequent cannot be of the form $b\backslash a^*$ or a/b^*, for the rules $\backslash R$ and $/R$ which yield conclusions with such goals have the property that their conclusion sequents, $T \vdash b\backslash a^*$ and $T \vdash a/b^*$, completely determine the form of their premise sequents: $b, T \vdash a^*$ and $T, b \vdash a^*$, respectively. So, c° must be (*i*) an asterisked atomic category at^*; or (*ii*) a non-asterisked atomic category at.

(*i*) If $c^\circ = at^*$, then $T^\circ \vdash c^\circ$ must have been obtained with the * rule in both π_1 and π_2. Hence $T^\circ = U, a, V, b, W$ and the situation is as follows in π_1 and π_2:

(29) $$\frac{U:U', a^*:u', V:V', b:v', W:W' \vdash at:\gamma'}{U:U', a:u', V:V', b:v', W:W' \vdash at^*:\gamma'}\,[*] \text{ and}$$

$$\frac{U:U'', a:u'', V:V'', b^*:v'', W:W'' \vdash at:\gamma''}{U:U'', a:u'', V:V'', b:v'', W:W'' \vdash at^*:\gamma''}\,[*]$$

It turns out that $\lambda U'\lambda u'\lambda V'\lambda v'\lambda W'.\gamma'$ and $\lambda U''\lambda u''\lambda V''\lambda v''\lambda W''.\gamma''$ are non-equivalent terms (see Fact 1 below).

- $\pi = \frac{\pi'}{\sigma}$, $\rho = \frac{\rho'}{\tau}$, $\pi' = \rho'$ and $\sigma = \tau$; or
- $\pi = \frac{\pi' \quad \pi''}{\sigma}$, $\rho = \frac{\rho' \quad \rho''}{\tau}$, $\pi' = \rho'$, $\pi'' = \rho''$ and $\sigma = \tau$.

Alternatively, we could enforce one particular assignment of different variables to each derivation π, for instance by requiring (*i*) that the sequence of axiom leafs of π equal $at_1{}^* : v_1 \vdash at_1 : v_1, \ldots, at_n{}^* : v_n \vdash at_n : v_n$, where for all i, $1 \leq i \leq n$: v_1 is the i-th variable of type TYPE(at_i); and (*ii*) that for all instances of $\backslash L$ and $/L$ in π the variable w be the i-th variable of type (TYPE(b),TYPE(a)) iff u is the i-th variable of type TYPE(a). Note that this would not restrict the semantic potential of the calculus, for it is easily checked that all π_1 and π_2 such that $\pi_1 = \pi_2$ in the sense just defined are semantically equivalent: if $\pi_1 = \pi_2$, then π_1 is a proof of $c_1 : v_1, \ldots, c_n : v_n \vdash c : \gamma$ and π_2 is a proof of $c_1 : v_1', \ldots, c_n : v_n' \vdash c : \gamma'$ such that $\lambda v_1 \ldots \lambda v_n.\gamma \equiv \lambda v_1' \ldots \lambda v_n'.\gamma'$ (and hence: $\lambda v_1 \ldots \lambda v_n.\gamma = \lambda v_1' \ldots \lambda v_n'.\gamma'$). The more liberal option has been chosen since it links up better with the alternative approaches to be discussed in the next section.

(*ii*) If $c^\circ = at$, then the left-hand side T° of $T^\circ \vdash c^\circ$ must contain one asterisked category d^*. Because $T^\circ \vdash c^\circ$ is not an axiom, d^* must be a non-atomic asterisked category a/b^* (or $b\backslash a^*$), and, accordingly, $T^\circ \vdash x^\circ$ must have been obtained using $/L$ (or $\backslash L$) in both π_1 and π_2. We only consider $/L$, since the $\backslash L$ case is analogous. Then d^* equals a/b^*; $T^\circ = U, a/b^*, V, W, X$, where W is a non-empty sequence of categories; and the situation is as follows in π_1 and π_2:

$$(30)\quad \frac{V:V' \vdash b^*:\beta' \qquad U:U', a^*:u', W:W', X:X' \vdash at:\gamma'}{U:U', a/b^*:w', V:V', W:W', X:X' \vdash at:\gamma'[u':=w'(\beta')]}\,[/L] \text{ and}$$

$$\frac{V:V'', W:W'' \vdash b^*:\beta'' \qquad U:U'', a^*:u'', X:X'' \vdash at:\gamma''}{U:U'', a/b^*:w'', V:V'', W:W'', X:X'' \vdash at:\gamma''[u'':=w''(\beta'')]}\,[/L]$$

The term $\lambda U'\lambda w'\lambda V'\lambda W'\lambda X'.\gamma'[u':=w'(\beta')]$ can be shown to be non-equivalent to $\lambda U''\lambda w''\lambda V''\lambda W''\lambda X''.\gamma''[u'':=w''(\beta'')]$ (see Fact 2 below).

Consequently, the intermediate conclusion $T^\circ \vdash c^\circ$ gets non-equivalent interpretations in π_1 and π_2.

Moreover, it can be shown that any non-equivalence of the terms which different proofs assign to some intermediate sequent is inherited by the terms assigned to the conclusions drawn from that sequent (see Fact 3 below). So, if the intermediate conclusion $T^\circ \vdash c^\circ$ has non-equivalent interpretations in π_1 and π_2, then $T \vdash c$, the final conclusion of both proofs, will receive non-equivalent interpretations in π_1 and π_2 as well. □

In footnote 3 it was observed that terms γ assigned to sequent goals in *Cut*-free **L** proofs are always in β-normal form, i.e., they do not have subterms of the form $[\lambda v.\gamma'](\gamma'')$. In the proof of Theorem 1 we have seen that the terms assigned to sequent goals in **L*** proofs constitute a subset of the terms assigned to sequent goals in *Cut*-free **L** proofs and, hence, are in β-normal form as well. However, they also meet the stronger requirement of being in what we will call '$\beta\bar{\eta}$-normal form' (as can be seen by an easy induction on the length of **L*** derivations):

> A term γ is in *$\bar{\eta}$-normal form* ($\bar{\eta}$-nf) iff every occurrence of a subterm $\gamma' \not\equiv \lambda v.\gamma''$ of functional type (a, b) is applied to some term γ''' in γ, i.e., it occurs as $\gamma'(\gamma''')$ in γ.[11]

[11] Let *η-expansion* ('$\bar{\eta}$-conversion') be defined as the replacement of a subterm occurrence

A term γ is in *$\beta\bar{\eta}$-nf* iff γ is in β-nf and γ is in $\bar{\eta}$-nf.

As was noted in footnote 4, equivalence in typed λ-calculus amounts to identity of $\beta\eta$-nfs, i.e., two terms γ and δ are equivalent ($\gamma = \delta$) iff their $\beta\eta$-nfs $\gamma^{\beta\eta}$ and $\delta^{\beta\eta}$ are syntactically identical modulo renaming bound variables ($\gamma^{\beta\eta} \equiv \delta^{\beta\eta}$). Together with the FACT mentioned in footnote 11, this yields that $\gamma = \delta$ iff $\gamma^{\beta\bar{\eta}} \equiv \delta^{\beta\bar{\eta}}$. Terms assigned to sequent goals in **L*** proofs are in $\beta\bar{\eta}$-nf, so they are equivalent just in case they are identical, a fact which will be exploited in the corollaries of the following lemma. Let the function $\#$ count the number of arguments of $c \in$ CAT, i.e., $\#(at) = 0$ and $\#(a/b) = \#(b\backslash a) = \#(a) + 1$.

Lemma:
If $U:U', c^*:w, V:V' \vdash at:\gamma$, then there are terms $\gamma_1, \ldots, \gamma_{\#(c)}$ such that $\gamma \equiv w(\gamma_{\#(c)})...(\gamma_1)$ and w does not occur in $\gamma_1, \ldots, \gamma_{\#(c)}$.

Proof: by induction on the degree of c.
• If $c \in$ ATOM, then $\#(c) = 0$ and $U:U', c^*:w, V:V' \vdash at:\gamma$ is an axiom sequent $at^*:w \vdash at:w$, so $\gamma \equiv w$.
• If $c = a/b$ (or $b\backslash a$), then $U:U', c^*:w, V:V' \vdash at:\gamma$ is the conclusion of a $/L$ (or, analogously, a $\backslash L$) inference:

$$\frac{V_1:V_1' \vdash b^*:\beta \qquad U:U', a^*:u, V_2:V_2' \vdash at:\gamma}{U:U', a/b^*:w, V_1:V_1', V_2:V_2' \vdash at:\gamma[u:=w(\beta)]}\,[/L]$$

Applying the induction hypothesis to γ, we know that there are terms $\gamma_1, \ldots, \gamma_{\#(a)}$ such that $\gamma \equiv u(\gamma_{\#(a)})...(\gamma_1)$ and the variable u does not occur in $\gamma_1, \ldots, \gamma_{\#(a)}$. Note that $\#(a/b) = \#(a)+1$, and that the term $\gamma[u:=w(\beta)]$ is identical to $w(\beta)(\gamma_{\#(a)})...(\gamma_1)$, where w, being different from the variables in β and γ, does not occur in $\gamma_1, \ldots, \gamma_{\#(a)}, \beta$. □

$\gamma' \not\equiv \lambda v.\gamma''$ of functional type (a,b) which is not applied to any term γ''' by $\lambda v.\gamma'(v)$, where v does not occur freely in γ'. One can show that η-expansion is strongly normalizing and Church-Rosser: for every term γ there is a unique term $\gamma^{\bar{\eta}}$ which is in $\bar{\eta}$-normal form, and which can be obtained from γ by performing a finite number of η-expansions (where renaming of bound variables is allowed). This is established by induction on the complexity of (the type of) γ, using the facts that (*i*) variables of non-functional (atomic) type are in $\bar{\eta}$-nf; (*ii*) η-expansion of variables of type (a,b) only introduces variables of less complex type; and (*iii*) η-expansion of complex terms $\lambda v.\gamma'$ and $\gamma'(\gamma'')$ only involves η-expansion of less complex terms. As η-expansion respects β-normality, we have the following FACT: for every γ and δ: $\gamma^{\beta\eta} \equiv \delta^{\beta\eta}$ iff $\gamma^{\beta\bar{\eta}} \equiv \delta^{\beta\bar{\eta}}$.

Let T' be a sequence of variables $v_1, \ldots, v_n$, and T a sequence of categories $c_1, \ldots, c_n$. We will say that T' is *of type* T iff for all i, $1 \leq i \leq n$: v_i is of type $\text{TYPE}(c_i)$. If $T' = v'_1, \ldots, v'_n$ is a sequence of different variables and $T'' = v''_1, \ldots, v''_n$ is a sequence of variables of the same type as T', then $\gamma[T' := T'']$ denotes the result of performing the simultaneous substitution $[v'_1 := v''_2, \ldots, v'_n := v''_n]$ in γ.

Note that for sequences T', T'' and T''' of different variables of the same type such that T' and T'' occur in γ' and γ'', respectively, and T''' occurs in γ' nor γ'', we have: $\lambda T'.\gamma' \equiv \lambda T''.\gamma''$ iff $\gamma'[T' := T'''] \equiv \gamma''[T'' := T''']$, since
- $\lambda T'''.[\gamma'[T' := T''']] \equiv \lambda T'''.[\gamma''[T'' := T''']]$ iff $\gamma'[T' := T'''] \equiv \gamma''[T'' := T''']$;
- $\lambda T'.\gamma' \equiv \lambda T'''.[\gamma'[T' := T''']]$ by renaming bound variables; and
- $\lambda T''.\gamma'' \equiv \lambda T'''.[\gamma''[T'' := T''']]$ by renaming bound variables.

Fact 1. Consider (29). By the above Lemma: $\gamma' \equiv u'(\gamma'_{\#(a)})\ldots(\gamma'_1)$ and $\gamma'' \equiv v''(\gamma''_{\#(b)})\ldots(\gamma''_1)$. Let $U''', u''', V''', v''', W'''$ be a sequence of different variables of type U, a, V, b, W which do not occur in γ' and γ''. Abbreviate U', u', V', v', W' as S'; U'', u'', V'', v'', W'' as S''; and $U''', u''', V''', v''', W'''$ as S'''. Then $\lambda S'.\gamma' \not\equiv \lambda S''.\gamma''$, since $\gamma'[S' := S'''] \equiv [u'(\gamma'_{\#(a)})\ldots(\gamma'_1)][S' := S'''] \equiv u'''(\gamma'_{\#(a)}[S' := S'''])\ldots(\gamma'_1[S' := S'''])$ $\not\equiv v'''(\gamma''_{\#(b)}[S'' := S'''])\ldots(\gamma''_1[S'' := S'''])$ $\equiv [v''(\gamma''_{\#(b)})\ldots(\gamma''_1)][S'' := S'''] \equiv \gamma''[S'' := S''']$. Therefore, $\lambda S'.\gamma' \neq \lambda S''.\gamma''$.

Fact 2. Consider (30). Again, by the Lemma: $\gamma' \equiv u'(\gamma'_{\#(a)})\ldots(\gamma'_1)$ and $\gamma'' \equiv u''(\gamma''_{\#(a)})\ldots(\gamma''_1)$, where u' and u'' do not occur in $\gamma'_1, \ldots, \gamma'_{\#(a)}$ and $\gamma''_1, \ldots, \gamma''_{\#(a)}$, respectively. Hence $\gamma'[u' := w'(\beta')] \equiv w'(\beta')(\gamma'_{\#(a)})\ldots(\gamma'_1)$ and $\gamma''[u'' := w''(\beta'')] \equiv w''(\beta'')(\gamma''_{\#(a)})\ldots(\gamma''_1)$. Since the sequence of categories W is non-empty, we have that $W = W_1, c, W_2$; $W' = W'_1, v', W'_2$; and $W'' = W''_1, v'', W''_2$. Note that for $1 \leq i \leq \#(a)$: v' occurs freely in some γ'_i but not in β', whereas v'' occurs freely in β'' but not in any γ''_i. Let $U''', w''', V''', W''', X'''$ be a sequence of different variables of type $U, a/b, V, W, X$ not occurring in $\gamma'[u' := w'(\beta')]$ and $\gamma''[u'' := w''(\beta')]$, and let $W''' = W'''_1, v''', W'''_2$. Denote U', w', V', W', X' by S'; U'', w'', V'', W'', X'' by S''; and $U''', w''', V''', W''', X'''$ by S'''.

We now have that $\lambda S'.[\gamma'[u' := w'(\beta')]] \not\equiv \lambda S''.[\gamma''[u'' := w''(\beta'')]]$, since $[\gamma'[u' := w'(\beta')]][S' := S'''] \not\equiv [\gamma''[u'' := w''(\beta'')]][S'' := S''']$:

- $[\gamma'[u' := w'(\beta')]][S' := S'''] \equiv [w'(\beta')(\gamma'_{\#(a)})\ldots(\gamma'_1)][S' := S'''] \equiv$ $w'''(\beta'[S' := S'''])(\gamma'_{\#(a)}[S' := S'''])\ldots(\gamma'_1[S' := S'''])$;

- $[\gamma''[u'':=w''(\beta'')]][S'':=S'''] \equiv [w''(\beta'')(\gamma''_{\#(a)})...(\gamma''_1)][S'':=S'''] \equiv w'''(\beta''[S'':=S'''])(\gamma''_{\#(a)}[S'':=S'''])...(\gamma''_1[S'':=S'''])$; and
- the variable v''' occupies the position of v' in $[\gamma'[u':=w'(\beta')]][S':=S''']$ and of v'' in $[\gamma''[u'':=w''(\beta'')]][S'':=S''']$, so v''' occurs freely in some $\gamma'_i[S':=S''']$ but not in $\beta'[S':=S''']$, and it occurs freely in $\beta''[S'':=S''']$ but not in any $\gamma''_i[S'':=S''']$.

Therefore: $\lambda S'.[\gamma'[u':=w'(\beta')]] \neq \lambda S''.[\gamma''[u'':=w''(\beta'')]]$.

Fact 3. Consider the following two inferences:

$$\frac{T:T' \vdash b^*:\beta' \qquad U:U', a^*:u', V:V' \vdash at:\gamma'}{U:U', a/b^*:w', T:T', V:V' \vdash at:\gamma'[u':=w'(\beta')]}\,[/L] \text{ and}$$

$$\frac{T:T'' \vdash b^*:\beta'' \qquad U:U'', a^*:u'', V:V'' \vdash at:\gamma''}{U:U'', a/b^*:w'', T:T'', V:V'' \vdash at:\gamma''[u'':=w''(\beta'')]}\,[/L]$$

By the Lemma: $\gamma' \equiv u'(\gamma'_{\#(a)})...(\gamma'_1)$, $\gamma'[u':=w'(\beta')] \equiv w'(\beta')(\gamma'_{\#(a)})...(\gamma'_1)$, $\gamma'' \equiv u''(\gamma''_{\#(a)})...(\gamma''_1)$ and $\gamma''[u'':=w''(\beta'')] \equiv w''(\beta'')(\gamma''_{\#(a)})...(\gamma''_1)$. Let U''', w''', T''', V''' be a sequence of different variables of type $U, a/b, T, V$ which do not occur in $\gamma'[u':=w'(\beta')]$ and $\gamma''[u'':=w''(\beta'')]$. Abbreviate U', w', T', V' as S'; U'', w'', T'', V'' as S''; and U''', w''', T''', V''' as S'''.

Now, suppose that $\lambda S'.[\gamma'[u':=w'(\beta')]] \equiv \lambda S''.[\gamma''[u'':=w''(\beta'')]]$. This entails that $[\gamma'[u':=w'(\beta')]][S':=S'''] \equiv [\gamma''[u'':=w''(\beta'')]][S'':=S''']$, that is: $[w'(\beta')(\gamma'_{\#(a)})...(\gamma'_1)][S':=S'''] \equiv [w''(\beta'')(\gamma''_{\#(a)})...(\gamma''_1)][S'':=S''']$.

So, $\beta'[S':=S'''] \equiv \beta''[S'':=S''']$ and for all i, $1 \leq i \leq \#(a)$: $\gamma'_i[S':=S'''] \equiv \gamma''_i[S'':=S''']$. Since the variables U', w', V' and U'', w'', V'' do not occur in β' and β'', respectively, we have that $\beta'[S':=S'''] \equiv \beta'[T':=T''']$ and $\beta''[S'':=S'''] \equiv \beta''[T'':=T''']$. So, $\lambda T'.\beta' \equiv \lambda T''.\beta''$. Moreover, since the variables w', T' and w'', T'' do not occur in any γ_i and γ''_i, respectively, we have that for all i: $\gamma'_i[S':=S'''] \equiv \gamma'_i[U', V':=U''', V''']$ and $\gamma''_i[S'':=S'''] \equiv \gamma''_i[U'', V'':=U''', V''']$. Finally, since u' and u'' are different from the variables in $\gamma'_1, \ldots, \gamma'_{\#(a)}$ and $\gamma''_1, \ldots, \gamma''_{\#(a)}$, respectively, we can conclude that $\lambda U'\lambda u'\lambda V'.u'(\gamma'_{\#(a)})...(\gamma'_1) \equiv \lambda U''\lambda u''\lambda V''.u''(\gamma''_{\#(a)})...(\gamma''_1)$.

Consequently, if $\lambda T'.\beta' \neq \lambda T''.\beta''$ or $\lambda U'\lambda u'\lambda V'.\gamma' \neq \lambda U''\lambda u''\lambda V''.\gamma''$, then $\lambda U'\lambda w'\lambda T'\lambda V'.[\gamma'[u':=w'(\beta')]] \neq \lambda U''\lambda w''\lambda T''\lambda V''.[\gamma''[u'':=w''(\beta'')]]$.

Furthermore, note (*i*) that the $\backslash L$ case is analogous; (*ii*) that the semantics of the * rule is identity; and (*iii*) that the cases $\backslash R$ and $/R$ are trival

as well. So, non-equivalence of terms assigned to some sequent is always inherited by the terms assigned to the conclusions drawn from that sequent.

Returning to the examples treated earlier, it turns out that the unambiguous sequent $c/c, c/c, c \vdash c^*$ is indeed derived only once in **L***, whereas it has two *Cut*-free derivations, (14) and (15), in **L**:

(31)
$$\dfrac{\dfrac{\dfrac{\dfrac{c^* \vdash c}{c \vdash c^*}[*] \quad c^* \vdash c}{c/c^*, c \vdash c}[/L]}{c/c, c \vdash c^*}[*] \quad c^* \vdash c}{c/c^*, c/c, c \vdash c}[/L]}{c/c, c/c, c \vdash c^*}[*]$$

Modulo * inferences, derivation (31) has the structure of (14). There is no **L*** proof with the structure of (15), for the main mmp of (15) contains two different active left-hand side categories.

The two-way ambiguous sequent $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$ which has the six *Cut*-free **L** derivations (16) through (21) is derived twice in **L***, cf. (32) and (33) below. These derivations are structurally similar and semantically equivalent to (16) and (19), respectively. The remaining *Cut*-free **L** derivations of the sequent do not have **L*** counterparts: (17) and (20) because of a non-atomic axiom instance $n\backslash s \vdash n\backslash s$; (18) due to a $\backslash R$ inference which yields the major premise of a $/L$ inference; and (21) on account of an mmp with two different active left-hand side categories.

(32)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n^* \vdash n}{n \vdash n^*}[*] \quad \dfrac{n^* \vdash n}{n \vdash n^*}[*] \quad s^* \vdash s}{n, n\backslash s^* \vdash s}[\backslash L]}{n, (n\backslash s)/n^*, n \vdash s}[/L]}{n, (n\backslash s)/n, n \vdash s^*}[*]}{n, (n\backslash s)/n \vdash s/n^*}[/R] \quad s^* \vdash s}{n, (n\backslash s)/n, (s/n)\backslash s^* \vdash s}[\backslash L]}{n, (n\backslash s)/n, (s/n)\backslash s \vdash s^*}[*]}{(n\backslash s)/n, (s/n)\backslash s^* \vdash n\backslash s^*}[\backslash R] \quad s^* \vdash s}{s/(n\backslash s)^*, (n\backslash s)/n, (s/n)\backslash s \vdash s}[/L]}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s^*}[*]$$

(33)

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n^* \vdash n}{n \vdash n^*}[*] \quad \dfrac{\dfrac{n^* \vdash n}{n \vdash n^*}[*] \quad s^* \vdash s}{n, n\backslash s^* \vdash s}[\backslash L]}{n, (n\backslash s)/n^*, n \vdash s}[/L]}{n, (n\backslash s)/n, n \vdash s^*}[*]}{(n\backslash s)/n, n \vdash n\backslash s^*}[\backslash R] \quad s^* \vdash s}{s/(n\backslash s)^*, (n\backslash s)/n, n \vdash s}[/L]}{s/(n\backslash s), (n\backslash s)/n, n \vdash s^*}[*]}{s/(n\backslash s), (n\backslash s)/n \vdash s/n^*}[/R] \quad s^* \vdash s}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s^* \vdash s}[\backslash L]}{s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s^*}[*]
$$

5 Partial Deduction and Proof Nets

In the present section we will discuss two alternative approaches to the problem of spurious ambiguity which were proposed in Moortgat (1990) and Roorda (1991). First, in Section 5.1, we give a formalization of the proposals in Moortgat (1990), and prove that the resulting proof representations 'summarize' **L*** derivations of sequents with atomic goal categories and are, hence, devoid of spurious ambiguity. Next, section 5.2 is devoted to a comparison of **L*** with the proof nets proposed in Roorda (1991). It is shown that every **L*** proof corresponds to a non-singleton *set* of proof nets.

5.1 Partial Deduction

Moortgat (1990) suggests that *partial deduction* offers a way of tackling the problem of spurious ambiguity for **L**. The basic idea of partial deduction is that theorem proving is preceded by a compilation of the *lexicon*, i.e., the set of categories which occur in the left-hand side of the sequents to be derived and interpreted. This set of categories serves as the basis for the generation of axioms and rules which are to replace the original axioms and rules of **L**.[12] The idea is that since they have more internal structure than the rules of **L**, the new rules restrict the derivational freedom of the original system, thus eliminating spurious ambiguity.

[12]Moortgat says that 'partial deduction of the lexicon adds *non-trivial* AXIOMS to the *trivial* axioms and logical rules' of **L**, such that the resulting system 'can do *without* the original logical rules' (1990, p. 388; small caps added, italics *sic*), but in the sequel we will see that the addition involves both axioms and rules.

Unfortunately, Moortgat (1990) does not contain a formal definition of partial deduction, let alone a proof that the method is sound and complete – i.e., that it eliminates all spurious ambiguity while preserving all **L**-derivable sequents and interpretations.

The paper only treats two examples, viz., the categories $(a_1\backslash a_2)/a_3$ and $(s/(n\backslash s))\backslash s$, where a_1, a_2, a_3, n and s are atomic.

The first order[13] category $(a_1\backslash a_2)/a_3$ gives rise to a single derived inference rule:[14]

$$(34)\quad \frac{T_2{:}T_2' \vdash a_3{:}\gamma_2 \qquad\qquad T_1{:}T_1' \vdash a_1{:}\gamma_1}{T_1{:}T_1', (a_1\backslash a_2)/a_3{:}v_0, T_2{:}T_2' \vdash a_2{:}v_0(\gamma_2)(\gamma_1)}\,[(a_1\backslash a_2)/a_3]$$

The higher order category $(s/(n\backslash s))\backslash s$, on the other hand, results in a *set* consisting of two derived inference rules. One corresponds to the category $(s/(n\backslash s))\backslash s$ itself, and the other to the (only) argument $n\backslash s$ of the (only) argument $s/(n\backslash s)$ of $(s/(n\backslash s))\backslash s$:

$$(35)\quad \frac{T_1{:}T_1', n\backslash s{:}v_1 \vdash s{:}\gamma_1}{T_1{:}T_1', (s/(n\backslash s))\backslash s{:}v_0 \vdash s{:}v_0(\lambda v_1.\gamma_1)}\,[(s/(n\backslash s))\backslash s]$$

$$\frac{T_1{:}T_1' \vdash n{:}\gamma_1}{T_1{:}T_1', n\backslash s{:}v_0 \vdash s{:}v_0(\gamma_1)}\,[n\backslash s]$$

We will show that a formal definition can be given which (*i*) is consistent with the examples treated by Moortgat; and (*ii*) indeed solves the problem of spurious ambiguity for sequents with atomic goal categories.

First, we define two functions from categories to sets of categories. For $c \in$ CAT, $A(c)$ is the set consisting of the arguments of c, while $A^A(c)$ is the set consisting of the arguments of the arguments of c:

$$A(at) = \emptyset;\ A(a/b) = A(b\backslash a) = \{b\} \cup A(a);$$
$$A^A(at) = \emptyset;\ A^A(a/b) = A^A(b\backslash a) = A(b) \cup A^A(a).$$

[13]The *order* of a category c, $o(c)$, is defined as follows: $o(c) = 0$ if $c \in$ ATOM; $o(a/b) = o(b\backslash a) = \max(o(a), o(b) + 1)$. For example: $o(n) = 0$, $o(n\backslash s) = 1$, $o(s/(n\backslash s)) = 2$ and $o((s/(n\backslash s))\backslash s) = 3$, whereas $o((s\backslash(n/s))/s) = 1$.

[14]Different from Moortgat (1990), but in keeping with our modification (10) of his semantics (5), the left-hand side categories $(a_1\backslash a_2)/a_3$, $(s/(n\backslash s))\backslash s$ and $n\backslash s$ are assigned a variable v_0 instead of a (possibly complex) term.

For a set of categories C, let C^+ be the set which includes C and contains, for $n \in \mathbb{N}$, any argument of (an argument of an argument of)^n an argument of a category $c \in C$. More formally:

> Let $C \subseteq$ CAT. C^+ is the smallest set such that
> (*i*) $C \subseteq C^+$, and (*ii*) for all $c \in C^+$: $A^A(c) \subseteq C^+$.

If C is the set of categories which appear in the left-hand side of sequents, then C^+ is the set of categories from which new axioms and rules are derived.

That C^+ is the required set can be seen from the examples (34) and (35). A derived inference rule associated with a category c deals with occurrences of c in the left-hand side of sequents: a category c consisting of n arguments $c_1, \ldots c_n$ and a (final) value at yields a derived inference rule consisting of a conclusion with left-hand side category c and goal category at, plus n premises with goal categories $c_1, \ldots c_n$ – that is: provided that $c_1, \ldots c_n$ are atomic categories, as in (34). If, as in (35), an argument category c_i of c itself consists of arguments $c'_1, \ldots c'_m$ and final value at', then the relevant premise has at' as its goal category, while the categories $c'_1, \ldots c'_m$ appear in its left-hand side. Now, the presence of arguments of arguments of c in the left-hand side of premises of the derived inference associated with c entails that these categories should be associated with a derived inference rule of their own.

For the example categories $(a_1\backslash a_2)/a_3$ and $(s/(n\backslash s))\backslash s$, the above definitions yield the following:

$A((a_1\backslash a_2)/a_3) = \{a_1, a_3\}$; $A^A((a_1\backslash a_2)/a_3) = \emptyset$;
so $\{(a_1\backslash a_2)/a_3\}^+ = \{(a_1\backslash a_2)/a_3\}$.
$A((s/(n\backslash s))\backslash s) = \{s/(n\backslash s)\}$; $A^A((s/(n\backslash s))\backslash s) = \{n\backslash s\}$;
$A(n\backslash s) = \{n\}$; $A^A(n\backslash s) = \emptyset$;
so $\{(s/(n\backslash s))\backslash s\}^+ = \{(s/(n\backslash s))\backslash s, n\backslash s\}$.

Next, let f be the following function from interpreted sequents to interpreted sequents:[15]

[15] Strictly speaking, f is a partial function, because terms assigned to goal categories a/b or $b\backslash a$ are not necessarily of the form $\lambda v.\gamma$.

Below we will also employ the following straightforward extension of f to interpreted sequents with asterisked goal categories:

$f(T{:}T' \vdash at^* : \gamma) = T{:}T' \vdash at^* : \gamma$;

$$f(T{:}T' \vdash at{:}\gamma) = T{:}T' \vdash at{:}\gamma$$
$$f(T{:}T' \vdash a/b{:}\lambda v.\gamma) = f(T{:}T', b{:}v \vdash a{:}\gamma); \text{ and}$$
$$f(T{:}T' \vdash b\backslash a{:}\lambda v.\gamma) = f(b{:}v, T{:}T' \vdash a{:}\gamma).$$

Syntactically, the function f peels off the argument categories from the goal category of the sequent to which it is applied. Semantically, it strips the term assigned to the goal category of its lambda abstractions. Together with the variables of the abstractions, these argument categories are added to the left-hand side of the sequent. An induction on $\#(c)$[16] shows that if a proof π of $T{:}T' \vdash c{:}\gamma$ is of the form indicated below, where $I_1, \ldots I_{\#(c)} \in \{\backslash R, /R\}$, then the conclusion of the subproof π' of π is the sequent $f(T{:}T' \vdash c{:}\gamma)$:

$$\dfrac{\dfrac{\pi'}{\vdots}\,[I_1]}{T{:}T' \vdash c{:}\gamma}\,[I_{\#(c)}]$$

The function g assigns an axiom to every atomic category at and a derived inference rule to every compound category $b\backslash a$ and a/b:

$$g(at) = at{:}v_0 \vdash at{:}v_0;$$
$$g(a/b) = \frac{f(T_{m+1}{:}T'_{m+1} \vdash b{:}\lambda v_1 \ldots \lambda v_{\#(b)}.\gamma_{m+1}) \qquad \sigma_m \quad \ldots \quad \sigma_1}{\mathbf{X}, a/b{:}v_0, T_{m+1}{:}T'_{m+1}, \mathbf{Y} \vdash at{:}v_0(\lambda v_1 \ldots \lambda v_{\#(b)}.\gamma_{m+1})(\delta_m)\ldots(\delta_1)}$$
and
$$g(b\backslash a) = \frac{f(T_{m+1}{:}T'_{m+1} \vdash b{:}\lambda v_1 \ldots \lambda v_{\#(b)}.\gamma_{m+1}) \qquad \sigma_m \quad \ldots \quad \sigma_1}{\mathbf{X}, T_{m+1}{:}T'_{m+1}, b\backslash a{:}v_0, \mathbf{Y} \vdash at{:}v_0(\lambda v_1 \ldots \lambda v_{\#(b)}.\gamma_{m+1})(\delta_m)\ldots(\delta_1)}$$

iff $g(a)$ consists of the conclusion $\mathbf{X}, a : v_0, \mathbf{Y} \vdash at : v_0(\delta_m)\ldots(\delta_1)$ and the premises $\sigma_m, \ldots, \sigma_1$. (Note that $m = \#(a)$.)

The expressions $T_1 : T'_1, \ldots, T_{\#(c)} : T'_{\#(c)}$ occurring in a rule $g(c)$ will range over non-empty finite sequences of pairs consisting of a category and a

---

$f(T{:}T' \vdash a/b^* : \lambda v.\gamma) = f(T{:}T', b{:}v \vdash a^*{:}\gamma)$; and
$f(T{:}T' \vdash b\backslash a^* : \lambda v.\gamma) = f(b{:}v, T{:}T' \vdash a^*{:}\gamma)$.

Note that since the terms assigned to goal categories in $\mathbf{L}^*$ derivations are in $\bar{\eta}$-nf (see section 4 above), the restriction of f to $\mathbf{L}^*$-derivable interpreted sequents is a total function.

[16]Recall that the function $\#$ counts the number of arguments of $c \in$ CAT: $\#(at) = 0$ and $\#(a/b) = \#(b\backslash a) = \#(a) + 1$.

variable.[17] As before, we will require that in the context of a proof all variables v_0 assigned to an axiom instance or introduced in the conclusion of a derived inference rule be different.

Let us see what these definitions amount to for the example categories discussed above. We start with the category $(a_1\backslash a_2)/a_3$. Since $\#(a_1) = \#(a_3) = 0$:

$$f(T_i{:}T_i' \vdash a_1{:}\lambda v_1 ... \lambda v_{\#(a_1)}.\gamma_i) = T_i{:}T_i' \vdash a_1{:}\gamma_i; \text{ and}$$
$$f(T_i{:}T_i' \vdash a_3{:}\lambda v_1 ... \lambda v_{\#(a_3)}.\gamma_i) = T_i{:}T_i' \vdash a_3{:}\gamma_i.$$

Since $g(a_2) = a_2{:}v_0 \vdash a_2{:}v_0$, we have:

$$g(a_1\backslash a_2) = \frac{T_1{:}T_1' \vdash a_1{:}\gamma_1}{T_1, a_1\backslash a_2{:}v_0 \vdash a_2{:}v_0(\gamma_1)}, \text{ and}$$

$$g((a_1\backslash a_2)/a_3) = \frac{T_2{:}T_2' \vdash a_3{:}\gamma_2 \qquad T_1{:}T_1' \vdash a_1{:}\gamma_1}{T_1{:}T_1', (a_1\backslash a_2)/a_3, T_2{:}T_2' \vdash a_2{:}v_0(\gamma_2)(\gamma_1)}.$$

Note that $g((a_1\backslash a_2)/a_3)$ is the same rule as (34). Now consider the second example, $(s/(n\backslash s))\backslash s$. Since $\#(s/(n\backslash s)) = 1$:

$$f(T_i{:}T_i' \vdash s/(n\backslash s){:}\lambda v_1 ... \lambda v_{\#(s/(n\backslash s))}.\gamma_i) = T_i{:}T_i', n\backslash s{:}v_1 \vdash s{:}\gamma_i.$$

Hence, and because $g(s) = s{:}v_0 \vdash s{:}v_0$:

$$g((s/(n\backslash s))\backslash s) = \frac{T_1{:}T_1', n\backslash s{:}v_1 \vdash s{:}\gamma_1}{T_1{:}T_1', (s/(n\backslash s))\backslash s{:}v_0 \vdash s{:}v_0(\lambda v_1.\gamma_1)}.$$

Since $\#(n) = 0$:

$$f(T_i{:}T_i' \vdash n{:}\lambda v_1 ... \lambda v_{\#(n)}.\gamma_i) = T_i{:}T_i' \vdash n{:}\gamma_i.$$

[17] Since the expressions $T_1 : T_1', \ldots, T_{\#(c)} : T_{\#(c)}'$ and $\gamma_1, \ldots, \gamma_{\#(c)}$ are distinguished by their subscripts, each $T_i{:}T_i'$ and γ_i can be instantiated independently. Strictly speaking, the variables $v_1, \ldots v_{\#(b_i)}$ and $v_1, \ldots v_{\#(b_j)}$ occurring in different premises σ_i and σ_j of a rule $g(c)$, as well as their counterparts in the conclusion of $g(c)$, should be distinguished for the same reason, for example in the following way:

$v_1^i, \ldots v_{\#(b_i)}^i$ and $v_1^j, \ldots v_{\#(b_j)}^j$.

Readability considerations, however, restrained us from doing so.

Moreover, $g(s) = s{:}v_0 \vdash s{:}v_0$, so that:

$$g(n\backslash s) = \quad \frac{T_1{:}T_1' \vdash n{:}\gamma_1}{T_1{:}T_1', n\backslash s{:}v_0 \vdash s{:}v_0(\gamma_1)}\,.$$

The rules $g((s/(n\backslash s))\backslash s)$ and $g(n\backslash s)$ are identical to the ones in (35). Thus, our formal definition is consistent with the examples of Moortgat (1990).

Call an interpreted sequent $c_1{:}v_1, \ldots c_n{:}v_n \vdash c{:}\gamma$ a **PD** ('Partial Deduction') theorem if and only if it is derivable using the axioms and rules in the set $\{g(c) \mid c \in \{c_1, \ldots, c_n\}^+\}$.

Notice that all **PD** theorems have atomic goal categories. We will show that there exists a surjection h from **L*** proofs of sequents $T{:}T' \vdash at^*{:}\gamma$ with asterisked atomic goal categories to **PD** proofs of $T{:}T' \vdash at{:}\gamma$.

On account of Theorem 1 (see section 3 above), this entails that partial deduction preserves all (interpretations of) **L**-derivable sequents with atomic goal categories.

Moreover, on account of Theorem 2 (see section 4 above), the existence of h entails that partial deduction is devoid of spurious ambiguity in that it derives every interpretation only once. The function h is defined as follows. Let $I_1, \ldots I_{\#(b)} \in \{\backslash R, /R\}$:

$$(a)\ h\left(\frac{at^*{:}w \vdash at{:}w}{at{:}w \vdash at^*{:}w}\,[*]\right) = at{:}w \vdash at{:}w;$$

$$(b)\ h\left(\frac{\dfrac{\dfrac{\pi'}{\vdots}\,[I_1]}{T{:}T' \vdash b^*{:}\beta}\,[I_{\#(b)}] \qquad \pi''}{\dfrac{U{:}U', a/b^*{:}w, T{:}T', V{:}V' \vdash at{:}\gamma[v{:=}w(\beta)]}{U{:}U', a/b{:}w, T{:}T', V{:}V' \vdash at^*{:}\gamma[v{:=}w(\beta)]}\,[*]}\,[/L]\right) =$$

$$\frac{h(\pi') \qquad \pi_n \qquad \ldots \qquad \pi_1}{U{:}U', a/b{:}w, T{:}T', V{:}V' \vdash at{:}\gamma[v{:=}w(\beta)]}\,[g(a/b)] \quad \text{iff}$$

$$h\left(\frac{\pi''}{U{:}U', a{:}v, V{:}V' \vdash at^*{:}\gamma}\,[*]\right) = \frac{\pi_n \qquad \ldots \qquad \pi_1}{U{:}U', a{:}v, V{:}V' \vdash at{:}\gamma}\,[g(a)];$$

$$(c)\quad h\left(\dfrac{\dfrac{\dfrac{\pi'}{\vdots}\,[I_1]}{T:T'\vdash b^*:\beta}\,[I_{\#(b)}] \qquad \pi''}{\dfrac{U:U',T:T',b\backslash a^*:w,V:V'\vdash at:\gamma[v:=w(\beta)]}{U:U',T:T',b\backslash a:w,V:V'\vdash at^*:\gamma[v:=w(\beta)]}\,[*]}\,[\backslash L]\right)=$$

$$\dfrac{h(\pi') \qquad \pi_n \qquad \cdots \qquad \pi_1}{U:U',T:T',b\backslash a:w,V:V'\vdash at:\gamma[v:=w(\beta)]}\,[g(b\backslash a)] \quad \text{iff}$$

$$h\left(\dfrac{\pi''}{U:U',a:v,V:V'\vdash at^*:\gamma}\,[*]\right)=\dfrac{\pi_n \qquad \cdots \qquad \pi_1}{U:U',a:v,V:V'\vdash at:\gamma}\,[g(a)].$$

On the level of proofs, the function h closely follows the pattern of the rule-building function g. Due to clause (a) of the definition of h, the most simple **L*** proofs π of sequents with asterisked atomic goal categories are mapped to **PD** proofs $h(\pi)$ which instantiate the value of g for atomic categories. Moreover, clause (b) maps more complex **L*** proofs π of sequents with asterisked atomic goal categories to **PD** proofs $h(\pi)$ in which the last rule applied is $g(a/b)$, where a/b is a category occurring in the left-hand side of the conclusion sequent of π. The $\#(a/b)$ subproofs of $h(\pi)$ which yield the premises of this application of the rule $g(a/b)$ are determined by two subproofs – π' and π'' – of π: π' is an **L*** proof of a sequent with asterisked atomic goal category, while π'' extended with * ('$\pi''+*$') is an **L*** proof of a sequent with asterisked atomic goal category such that $h(\pi''+*)$ is a **PD** proof in which the last rule applied is $g(a)$. Now, given the way in which $g(a/b)$ is defined (viz., in terms of $g(a)$), the conclusion of $h(\pi')$ plus the premises of $g(a)$ in $h(\pi''+*)$ can serve as the premises of an application of $g(a/b)$ which infers the non-asterisked counterpart of the conclusion of π. (An isomorphic story can be told about clause (c).)

We will now prove the following facts: (1) for every **L*** proof π of a sequent $Z:Z'\vdash at^*:\delta$ with asterisked atomic goal category, $h(\pi)$ is a **PD** proof of $Z:Z'\vdash at:\delta$; and (2) for every **PD** proof π, there is an **L*** proof π^* such that $h(\pi^*)=\pi$.

(1) That for every **L*** proof π of a sequent $Z:Z'\vdash at^*:\delta$ with asterisked atomic goal, $h(\pi)$ is a **PD** proof of $Z:Z'\vdash at:\delta$ is shown by induction on d, the degree of the conclusion $Z:Z'\vdash at^*:\delta$ of π. Note that in view of its goal category at^*, the sequent $Z:Z'\vdash at^*:\delta$, i.e., $U:U',c:w,W:W'\vdash at^*:\delta$,

must have been derived by the * rule from $U:U', c^*:w, W:W' \vdash at:\delta$.
• $d = 0$. Then the sequent $U:U', c^*:w, W:W' \vdash at:\delta$ must be an L* axiom $at^*:w \vdash at:w$. This case is covered by clause (a) of the definition of h which states that $h(\pi) = at:w \vdash at:w$, i.e., an application of $g(at)$.
• $d > 0$. There are two possibilities: $c = a/b$ or $c = b\backslash a$. We only consider the former case. (The $b\backslash a$ case is analogous.) If $c = a/b$, then $W:W' = T:T', V:V'$ and $\delta \equiv \gamma[v:=w(\beta)]$, because $U:U', c^*:w, W:W' \vdash at:\delta$, i.e., $U:U', a/b^*:w, T:T', V:V' \vdash at:\gamma[v:=w(\beta)]$, must have been inferred by $/L$ from the premises (a) $T:T' \vdash b^*:\beta$ and (b) $U:U', a^*:v, V:V' \vdash at:\gamma$.

(a) As for the subproof of $T:T' \vdash b^*:\beta$: such a sequent must have been inferred using * (if $b = at$), $\backslash R$ (if $b = b''\backslash b'$) or $/R$ (if $b = b'/b''$). Besides, premises of $\backslash R$ and $/R$ inferences have asterisked goal categories as well. So this subproof must have the following form, where $I_1, \ldots I_{\#(b)} \in \{\backslash R, /R\}$:

$$\pi'\left[\begin{array}{c} \dfrac{\pi_0}{\sigma_0}\,[*] \\ \vdots \\ \dfrac{}{T:T' \vdash b^*:\beta}\,[I_{\#(b)}] \end{array}\right.\ [I_1]$$

Note that since every I_i $(1 \leq i \leq \#(b))$ adds an abstraction λu_i to the term assigned to the goal category of its premise, the term β must be of the form $\lambda u_1 ... \lambda u_{\#(b)}.\beta'$. So, using the observation concerning f made above, we know that σ_0 is the sequent $f(T:T' \vdash b^*:\lambda u_1 \ldots \lambda u_{\#(b)}.\beta')$. This sequent has an asterisked atomic goal and its degree is smaller than the degree of the conclusion of π. Hence we can apply the induction hypothesis to π' and infer that $h(\pi')$ is a **PD** proof of $f(T:T' \vdash b:\lambda u_1 \ldots \lambda u_{\#(b)}.\beta')$ – i.e., $f(T:T' \vdash b^*:\lambda u_1 \ldots \lambda u_{\#(b)}.\beta')$ minus asterisk.

(b) As for the subproof π'' of $U:U', a^*:v, V:V' \vdash at:\gamma$: note that we can apply the Lemma of section 4 above: $\gamma \equiv v(\gamma_{\#(a)})...(\gamma_1)$, where v does not occur in $\gamma_1, \ldots, \gamma_{\#(a)}$. Moreover, the derivation π''' consisting of π'' plus an additional * inference is an **L*** proof of a sequent with an asterisked atomic goal category:

$$\pi''' = \frac{\pi''}{U:U', a:v, V:V' \vdash at^*:\gamma}\,[*]$$

The degree of $U:U', a:v, V:V' \vdash at^*:\gamma$ is smaller than the degree of the conclusion of π. Hence $h(\pi''')$ is a **PD** proof of $U:U', a:v, V:V' \vdash at:\gamma$ by the induction hypothesis. Since $\gamma \equiv v(\gamma_{\#(a)})...(\gamma_1)$, we know that $g(a)$ must

be the last inference of $h(\pi''')$, a proof which, therefore, has the following form:

$$h(\pi''') = \frac{\pi_{\#(a)} \quad \cdots \quad \pi_1}{U:U', a:v, V:V' \vdash at:v(\gamma_{\#(a)})...(\gamma_1)}\ [g(a)]$$

Now, according to clause (b) of the definition of h,

$$h(\pi) = \frac{h(\pi') \quad \pi_{\#(a)} \quad \cdots \quad \pi_1}{U:U', a/b:w, T:T', V:V' \vdash at:\gamma[v:=w(\beta)]}\ [g(a/b)],$$

which is indeed a **PD** proof, for recall that

$$g(a/b) = \frac{f(T_{m+1}:T'_{m+1} \vdash b:\lambda v_1...\lambda v_{\#(b)}.\gamma_{m+1}) \quad \sigma_{\#(a)} \quad \cdots \quad \sigma_1}{\mathbf{X}, a/b:v_0, T_{m+1}:T'_{m+1}, \mathbf{Y} \vdash at:v_0(\lambda v_1...\lambda v_{\#(b)}.\gamma_{m+1})(\delta_{\#(a)})...(\delta_1)},$$

where $\sigma_{\#(a)}, \ldots, \sigma_1$ and $\mathbf{X}, a:v_0, \mathbf{Y} \vdash at:v_0(\delta_{\#(a)})...(\delta_1)$ are the premises and conclusion of $g(a)$, and note that

- apparently, the conclusions of the subproofs $\pi_{\#(a)}, \ldots, \pi_1$ match the respective premises $\sigma_{\#(a)}, \ldots, \sigma_1$ of the rule $g(a)$ in such a way that $at:v(\gamma_{\#(a)})...(\gamma_1)$, $U:U'$, $a:v$ and $V:V'$ match $at:v_0(\delta_{\#(a)})...(\delta_1)$, $\mathbf{X}$, $a:v_0$ and $\mathbf{Y}$, respectively; and
- obviously, the conclusion $f(T:T' \vdash b:\lambda u_1...\lambda u_{\#(b)}.\beta')$ of $h(\pi')$ matches the first premise $f(T_{m+1}:T'_{m+1} \vdash b:\lambda v_1...\lambda v_{\#(b)}.\gamma_{m+1})$ of the rule $g(a/b)$ in such a way that $T:T'$ and $\lambda u_1...\lambda u_{\#(b)}.\beta'$ match $T_{m+1}:T'_{m+1}$ and $\lambda v_1...\lambda v_{\#(b)}.\gamma_{m+1}$, respectively.

But this means that the conclusion of $g(a/b)$ is matched by the sequent $U:U', a/b:w, T:T', V:V' \vdash at:w(\lambda u_1...\lambda u_{\#(b)}.\beta')(\gamma_{\#(a)})...(\gamma_1)$. Note, finally, that $w(\lambda u_1 \ldots \lambda u_{\#(b)}.\beta')(\gamma_{\#(a)})...(\gamma_1) \equiv \gamma[v:=w(\beta)]$, because $\gamma \equiv v(\gamma_{\#(a)})...(\gamma_1)$, the variable v does not occur in $\gamma_1, \ldots, \gamma_{\#(a)}$, and $\beta \equiv \lambda u_1 \ldots \lambda u_{\#(b)}.\beta'$. □

(2) We show that for every **PD** proof π there is an **L*** proof π^* such that $h(\pi^*) = \pi$ by induction on d, the degree of the conclusion $Z:Z' \vdash at:\delta$ of the proof π.

• $d = 0$. Then Z only contains atomic categories, so for $c \in$ ATOM: π must match $g(c) = at:v_0 \vdash at:v_0$. Consequently, $\pi = at:v \vdash at:v$, and:

$$at{:}v \vdash at{:}v = h\left(\frac{at^*{:}v \vdash at{:}v}{at{:}v \vdash at^*{:}v}\,[^*]\right)$$

• $d > 0$. Then the conclusion of π must have been obtained by a derived inference rule $g(c)$ such that the degree of c is greater than 0. So:

$$\pi = \frac{\pi_{\#(a)+1} \quad \pi_{\#(a)} \quad \cdots \quad \pi_1}{U{:}U', b\backslash a{:}w, W{:}W' \vdash at{:}\delta}\,[g(b\backslash a)], \text{ or}$$

$$\pi = \frac{\pi_{\#(a)+1} \quad \pi_{\#(a)} \quad \cdots \quad \pi_1}{U{:}U', a/b{:}w, W{:}W' \vdash at{:}\delta}\,[g(a/b)],$$

We will only consider the case of $g(a/b)$. (The case of $g(b\backslash a)$ is analogous.) Then $W{:}W' = T{:}T', V{:}V'$ and $\gamma \equiv w(\lambda u_1 ... \lambda u_{\#(b)}.\beta')(\gamma_{\#(a)}) ... (\gamma_1)$, where (i) the conclusion of $\pi_{\#(a)+1}$ is $f(T{:}T' \vdash b{:}\lambda u_1 ... \lambda u_{\#(b)}.\beta')$; and (ii) the conclusions of the subproofs $\pi_{\#(a)}, \ldots, \pi_1$ match the respective premises $\sigma_{\#(a)}, \ldots, \sigma_1$ of the rule $g(a)$ in such a way that application of that rule to these sequents yields $U{:}U', a{:}v, V{:}V' \vdash at{:}v(\gamma_{\#(a)}) ... (\gamma_1)$.

Since the degree of the sequents $f(T : T' \vdash b : \lambda u_1 ... \lambda u_{\#(b)}.\beta')$ and $U : U', a : v, V : V' \vdash at : v(\gamma_{\#(a)}) ... (\gamma_1)$ is smaller than the degree of the conclusion of π, we can apply the induction hypothesis to $\pi_{\#(a)+1}$ and $\rho =$

$$\frac{\pi_{\#(a)} \quad \cdots \quad \pi_1}{U{:}U', a{:}v, V{:}V' \vdash at{:}v(\gamma_{\#(a)}) ... (\gamma_1)}\,[g(a)],$$

and conclude that there are $\mathbf{L}^*$ proofs ${\pi_{\#(a)+1}}^*$ and ρ^* such that $h({\pi_{\#(a)+1}}^*) = \pi_{\#(a)+1}$ and $h(\rho^*) = \rho$.

Because $\pi_{\#(a)+1}$ and ρ are proofs of $f(T : T \vdash b : \lambda u_1 ... \lambda u_{\#(b)}.\beta')$ and $U : U', a : v, V : V' \vdash at : v(\gamma_{\#(a)}) ... (\gamma_1)$, respectively, we know that ${\pi_{\#(a)+1}}^*$ and ρ^* are proofs of the asterisked sequents $f(T : T \vdash b^* : \lambda u_1 ... \lambda u_{\#(b)}.\beta')$ and $U{:}U', a{:}v, V{:}V' \vdash at^*{:}v(\gamma_{\#(a)}) ... (\gamma_1)$, respectively.

As for ${\pi_{\#(a)+1}}^*$: it can be shown by induction on $\#(c)$ that any $\mathbf{L}^*$ proof π^* of $f(T : T \vdash c^* : \lambda u_1 ... \lambda u_{\#(c)}.\beta')$ can be extended to an $\mathbf{L}^*$ proof ${\pi^*}^+$ of $T : T \vdash c^* : \lambda u_1 ... \lambda u_{\#(c)}.\beta'$ by adding $\#(c)$ $\backslash R$ and $/R$ inferences to π^*. Hence there exists such an $\mathbf{L}^*$ proof ${\pi_{\#(a)+1}}^{*+}$ of $T{:}T \vdash b^*{:}\lambda u_1 ... \lambda u_{\#(b)}.\beta'$.

As for ρ^*: since $U : U', a : v, V : V' \vdash at^* : v(\gamma_{\#(a)}) ... (\gamma_1)$, the conclusion of ρ^*, has an asterisked atomic goal which is assigned a term with leading

variable v, we know (by the Lemma of section 4) that

$$\rho^* = \dfrac{\rho^{*\prime}}{U:U', a:v, V:V' \vdash at^*:v(\gamma_{\#(a)})...(\gamma_1)}\,[*],$$

where $U:U', a^*:v, V:V' \vdash at:v(\gamma_{\#(a)})...(\gamma_1)$ is the conclusion of $\rho^{*\prime}$. Hence π^*, the proof below, is an $\mathbf{L}^*$ derivation, where $I_1, \ldots, I_{\#(b)} \in \{\backslash R, /R\}$:

$$\dfrac{\dfrac{\left.\dfrac{\dfrac{\pi_{\#(a)+1}{}^*}{\vdots}\,[I_1]}{T:T' \vdash b^*:\lambda u_1...\lambda u_{\#(b)}.\beta'}\,[I_{\#(b)}]\right]\pi_{\#(a)+1}{}^{*+} \qquad \rho^{*\prime}}{U:U', a/b^*:w, T:T', V:V' \vdash at:w(\lambda u_1...\lambda u_{\#(b)}.\beta')(\gamma_{\#(a)})...(\gamma_1)}\,[/L]}{U:U', a/b:w, T:T', V:V' \vdash at^*:w(\lambda u_1...\lambda u_{\#(b)}.\beta')(\gamma_{\#(a)})...(\gamma_1)}\,[*]$$

Finally, because $h(\rho^*) = \rho =$

$$\dfrac{\pi_{\#(a)} \qquad \cdots \qquad \pi_1}{U:U', a:v, V:V' \vdash at:v(\gamma_{\#(a)})...(\gamma_1)}\,[g(a)],$$

clause (b) of the definition of h tells us that $h(\pi^*) =$

$$\dfrac{h(\pi_{\#(a)+1}{}^*) \qquad \pi_{\#(a)} \qquad \cdots \qquad \pi_1}{U:U', a/b:w, T:T', V:V' \vdash at:w(\lambda u_1...\lambda u_{\#(b)}.\beta')(\gamma_{\#(a)})...(\gamma_1)}\,[g(a/b)]$$

And because $h(\pi_{\#(a)+1}{}^*) = \pi_{\#(a)+1}$, we have that $h(\pi^*) =$

$$\dfrac{\pi_{\#(a)+1} \qquad \pi_{\#(a)} \qquad \cdots \qquad \pi_1}{U:U', a/b:w, T:T', V:V' \vdash at:w(\lambda u_1...\lambda u_{\#(b)}.\beta')(\gamma_{\#(a)})...(\gamma_1)}\,[g(a/b)],$$

which is nothing else but π. □

By way of illustration, we return to the sample sequents $c/c, c/c, c \vdash c$ and $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$ treated in the previous sections. The categories c/c and c generate one rule and one axiom:

$$\dfrac{T_1:T_1' \vdash c:\gamma_1}{c/c:v_0, T_1:T_1' \vdash c:v_0(\gamma_1)}\,[g(c/c)]$$

$$c:v_0 \vdash c:v_0 \quad [g(c)]$$

These yield one **PD** derivation of $c/c, c/c, c \vdash c$, viz., (36). Observe that (36) = h(31):

$$(36)\quad \dfrac{\dfrac{c{:}x \vdash c{:}x}{c/c{:}y, c{:}x \vdash c{:}y(x)}\,[g(c/c)]}{c/c{:}z, c/c{:}y, c{:}x \vdash c{:}z(y(x))}\,[g(c/c)]$$

The categories $s/(n\backslash s), (s/n)\backslash s$ and $(n\backslash s)/n$ generate the following rules and axiom:

$$\dfrac{n{:}v_1, T_1{:}T_1' \vdash s{:}\gamma_1}{s/(n\backslash s){:}v_0, T_1{:}T_1' \vdash c{:}v_0(\lambda v_1.\gamma_1)}\,[g(s/(n\backslash s))]$$

$$\dfrac{T_1{:}T_1', n{:}v_1 \vdash s{:}\gamma_1}{T_1{:}T_1', (s/n)\backslash s{:}v_0 \vdash c{:}v_0(\lambda v_1.\gamma_1)}\,[g((s/n)\backslash s)]$$

$$n{:}v_0 \vdash n{:}v_0 \quad [g(n)]$$

$$\dfrac{T_2{:}T_2' \vdash n{:}\gamma_2 \qquad T_1{:}T_1' \vdash n{:}\gamma_1}{T_1{:}T_1', (n\backslash s)/n{:}v_0, T_2{:}T_2' \vdash c{:}v_0(\gamma_2)(\gamma_1)}\,[g((n\backslash s)/n)]$$

These yield two proofs of $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$, viz., (37) and (38), where (37) = h(32) and (38) = h(33):

$$(37)\quad \dfrac{\dfrac{\dfrac{n{:}w' \vdash n{:}w' \qquad n{:}u' \vdash n{:}u'}{n{:}u', (n\backslash s)/n{:}v, n{:}w' \vdash s{:}v(w')(u')}\,[g((n\backslash s)/n)]}{n{:}u', (n\backslash s)/n{:}v, (s/n)\backslash s{:}w \vdash s{:}w(\lambda w'.v(w')(u'))}\,[g((s/n)\backslash s)]}{s/(n\backslash s){:}u, (n\backslash s)/n{:}v, (s/n)\backslash s{:}w \vdash s{:}u(\lambda u'.w(\lambda w'.v(w')(u')))}\,[g(s/(n\backslash s))]$$

$$(38)\quad \dfrac{\dfrac{\dfrac{n{:}w' \vdash n{:}w' \qquad n{:}u' \vdash n{:}u'}{n{:}u', (n\backslash s)/n{:}v, n{:}w' \vdash s{:}v(w')(u')}\,[g((n\backslash s)/n)]}{s/(n\backslash s){:}u, (n\backslash s)/n{:}v, n{:}w' \vdash s{:}u(\lambda u'.v(w')(u'))}\,[g(s/(n\backslash s))]}{s/(n\backslash s){:}u, (n\backslash s)/n{:}v, (s/n)\backslash s{:}w \vdash s{:}w(\lambda w'.u(\lambda u'.v(w')(u')))}\,[g((s/n)\backslash s)]$$

5.2 Proof Nets

Roorda (1991) introduces so-called *proof nets* – which were invented for linear logic by Girard (1987) – for the Lambek calculus. Proof nets are proposed as a solution to the problem of spurious ambiguity, in that they replace 'inessential' order of rule applications in sequent proofs by 'parallellism'.

In the present section we will go into the details of this proposal. We start with the definition of proof nets given in Roorda (1991), which is listed in (*a*) through (*g*) below. Next, an alternative, equivalent definition is presented in (*a'*) through (*h'*). While Roorda's definition recursively introduces connectives, the alternative one proceeds by adding links. This different set-up has two advantages. First, it allows a direct, inductive definition of the semantic interpretation of proof nets – see the definition given in (*a''*) through (*h''*). And second, it brings out the correspondence between applying rules in **L*** and adding links to proof nets: there will turn out to be a one-many relationship between **L*** proofs and proof nets.

Basically, a proof net is a structure of linked signed categories. For the product-free Lambek calculus, the following links are distinguished:

(39) $\dfrac{}{\overset{-}{at}\ \overset{+}{at}}\,ax$ $\dfrac{}{\overset{+}{at}\ \overset{-}{at}}\,ax$ $\dfrac{\overset{+}{a}\quad\overset{-}{b}}{\overset{+}{b\backslash a}}\backslash 1$ $\dfrac{\overset{-}{b}\quad\overset{+}{a}}{\overset{+}{a/b}}/1$ $\dfrac{\overset{+}{b}\quad\overset{-}{a}}{\overset{-}{b\backslash a}}\backslash 2$ $\dfrac{\overset{-}{a}\quad\overset{+}{b}}{\overset{-}{a/b}}/2$

Below, the expressions X, Y and Z will denote sequences of signed categories, and $tc(C)$ will be used for the sequence of so-called *terminal* (signed) *categories* of a proof net C.

PN, the set of proof nets, is the smallest set satisfying the clauses (*a*) through (*g*) below, which also specify the sequence $tc(C)$ for each $C \in$ PN:[18]

[18]This definition is based on the definition in Roorda (1991, p. 30), but there are two differences: (*i*) the condition 'X or Y is non-empty' has been added to clause (*b*) and (*c*), to the effect that every proof net has at least one terminal category signed '$-$'. This mirrors the 'non-empty antecedent property' of **L**, i.e., the fact that the left-hand side of **L**-derivable sequents is non-empty, a property which was effectuated in (4) above by requiring that T denote a non-empty sequence of categories (Roorda (1991, p. 37) captures this aspect by a 'proof net condition' on the assignment of lambda terms); and (*ii*) anticipating the semantic interpretation of proof nets, the various cases collapsed by Roorda have been disentangled. (Note, for that matter, that Roorda collapses too much: his clauses **2A** and **2B** – which comprise the clauses (*d*) and (*g*), and (*e*) and (*f*), respectively – have the same condition: 'if β and γ are proof nets with terminal formulas tA resp. uBv' (t, u, v and A, B range over sequences of signed categories and signed categories,

(*a*) $\overline{\overset{-}{at}\ \overset{+}{at}} \in$ PN and $\overline{\overset{+}{at}\ \overset{-}{at}} \in$ PN;
$tc(\overline{\overset{-}{at}\ \overset{+}{at}}) = \overset{-}{at},\overset{+}{at}$ and $tc(\overline{\overset{+}{at}\ \overset{-}{at}}) = \overset{+}{at},\overset{-}{at}$;

(*b*) if $A \in$ PN and $tc(A) = X, \overset{+}{a}, \overset{-}{b}, Y$ and X or Y is non-empty,
then $C \in$ PN, where C results from applying $\backslash 1$ to $\overset{+}{a}$ and $\overset{-}{b}$;
$tc(C) = X, \overset{+}{b\backslash a}, Y$;

(*c*) if $A \in$ PN and $tc(A) = X, \overset{-}{b}, \overset{+}{a}, Y$ and X or Y is non-empty,
then $C \in$ PN, where C results from applying $/1$ to $\overset{-}{b}$ and $\overset{+}{a}$;
$tc(C) = X, \overset{+}{a/b}, Y$;

(*d*) if $A \in$ PN and $tc(A) = X, \overset{-}{a}, Y$ and $B \in$ PN and $tc(B) = Z, \overset{+}{b}$,
then $C \in$ PN, where C results from applying $\backslash 2$ to $\overset{+}{b}$ and $\overset{-}{a}$;
$tc(C) = X, Z, \overset{-}{b\backslash a}, Y$;

(*e*) if $A \in$ PN and $tc(A) = X, \overset{+}{b}, Y$ and $B \in$ PN and $tc(B) = \overset{-}{a}, Z$,
then $C \in$ PN, where C results from applying $\backslash 2$ to $\overset{+}{b}$ and $\overset{-}{a}$;
$tc(C) = X, \overset{-}{b\backslash a}, Z, Y$;

(*f*) if $A \in$ PN and $tc(A) = X, \overset{-}{a}, Y$ and $B \in$ PN and $tc(B) = \overset{+}{b}, Z$,
then $C \in$ PN, where C results from applying $/2$ to $\overset{-}{a}$ and $\overset{+}{b}$;
$tc(C) = X, \overset{-}{a/b}, Z, Y$;

(*g*) if $A \in$ PN and $tc(A) = X, \overset{+}{b}, Y$ and $B \in$ PN and $tc(B) = Z, \overset{-}{a}$,
then $C \in$ PN, where C results from applying $/2$ to $\overset{-}{a}$ and $\overset{+}{b}$;
$tc(C) = X, Z, \overset{-}{a/b}, Y$.

The proof nets defined in clause (*a*) correspond to the axioms of **L** (where observation **A** of section 3 above – entailing the possibility of having only axiom sequents with atomic categories – has been incorporated), while the clauses (*b*), (*c*), (*d*) + (*e*), and (*f*) + (*g*) are the proof net counterparts of the **L** rules $\backslash R$, $/R$, $\backslash L$ and $/L$, respectively.

On account of the clauses (*d*), (*e*), (*f*) and (*g*), two separate proof nets

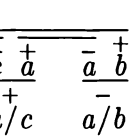
respectively). Since one has to be 'strict on order and adjacency of premises in proof nets' (1991, p. 30), however, this will not work for clause **2B**, which needs a separate condition, viz., 'if β and γ are proof nets with terminal formulas At resp. uBv'.)

can be combined into one: application of $\backslash 2$ and $/2$ has the effect of *inserting* proof net B into proof net A.

Note that for every proof net C, the sequence $tc(C)$ consists of exactly one category signed '+' and at least one category signed '−'. We will call the unique positively signed category in $tc(C)$ the *goal* of C, which will be denoted by $go(C)$. A proof net C *corresponds* to a sequent $T \vdash c$ iff

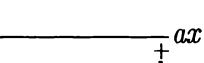
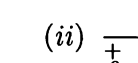
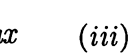
$$T = c_1, \ldots, c_i, c_{i+1}, \ldots, c_n \text{ and } tc(C) = c_{i+1}^{-}, \ldots, c_n^{-}, c^{+}, c_1^{-}, \ldots, c_i^{-}.$$

By way of illustration of the definition, consider proof net (40), which corresponds to the **L**-derivable sequent $a/b, b/c \vdash a/c$; $tc(40)$ is the sequence listed in (40′).

(40) 

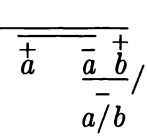
(40′) $b^{-}/c, a^{+}/c, a^{-}/b$

Proof net (40) can be built up in two ways from the basic proof nets in (41), which have the sequences of terminal categories listed in (42):

(41) (*i*)  (*ii*)  (*iii*) 

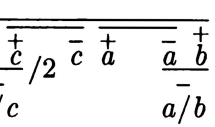
(42) $tc(i) = b^{-}, b^{+}$; $tc(ii) = c^{+}, c^{-}$; $tc(iii) = a^{+}, a^{-}$.

A first possibility is that we add a $/2$ link to a^{-} in (*iii*) and b^{+} in (*i*), due to clause (*g*) (where (*i*) and (*iii*) instantiate A and B, respectively). Note that (*iii*) is inserted into (*i*). The resulting proof net (43) has the terminal categories indicated in (43′):

(43) 

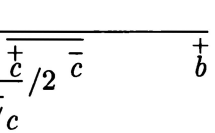
(43′) $b^{-}, a^{+}, a^{-}/b$

Next, we can add a $/2$ link to b^{-} in (43) and c^{+} in (*ii*), due to clause (*f*) (where (43) and (*ii*) instantiate A and B, respectively): (*ii*) is inserted into (43). The resulting proof net (44) has the terminal categories indicated in (44′):

(44) 

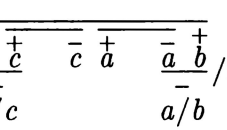
(44′) $\overset{-}{b}/c, \overset{-}{c}, \overset{+}{a}, a/\overset{-}{b}$

Secondly, we may add a /2 link to $\overset{-}{b}$ in (*i*) and $\overset{+}{c}$ in (*ii*), due to clause (*f*) (where (*i*) and (*ii*) instantiate A and B, respectively): (*ii*) is inserted into (*i*). The resulting proof net (45) has the terminal categories indicated in (45′):

(45) 

(45′) $\overset{-}{b}/c, \overset{-}{c}, \overset{+}{b}$

Next, we can add a /2 link to $\overset{-}{a}$ in (*iii*) and $\overset{+}{b}$ in (45), due to clause (*g*) (where (45) and (*iii*) instantiate A and B, respectively): (*iii*) is inserted into (45). The resulting proof net (46) is identical to (44):

(46) 

(46′) $\overset{-}{b}/c, \overset{-}{c}, \overset{+}{a}, a/\overset{-}{b}$

Finally, the addition of a /1 link (by clause (*c*)) to the categories $\overset{-}{c}$ and $\overset{+}{a}$ in proof net (44) (= (46)) yields proof net (40).

The above definition of the set PN is equivalent to the following one (in the sense that $\text{PN} = \text{PN}_2$), in which *two* sets, PN_1 and PN_2, are defined as the smallest sets such that:

(*a*′) $\overset{-}{at}\ \overset{+}{at} \in \text{PN}_1$ and $\overset{+}{at}\ \overset{-}{at} \in \text{PN}_1$;
$tc(\overset{-}{at}\ \overset{+}{at}) = \overset{-}{at}, \overset{+}{at}$ and $tc(\overset{+}{at}\ \overset{-}{at}) = \overset{+}{at}, \overset{-}{at}$;

(*b*′) if $A \in \text{PN}_2$ and $tc(A) = X, \overset{+}{a}, \overset{-}{b}, Y$ and X or Y is non-empty,
then $C \in \text{PN}_2$, where C results from applying $\backslash 1$ to $\overset{+}{a}$ and $\overset{-}{b}$;
$tc(C) = X, \overset{+}{b}\backslash a, Y$;

(*c*′) if $A \in \text{PN}_2$ and $tc(A) = X, \overset{-}{b}, \overset{+}{a}, Y$ and X or Y is non-empty,

then $C \in \text{PN}_2$, where C results from applying /1 to $\bar{b}$ and $\overset{+}{a}$;
$tc(C) = X, \overset{+}{a}/b, Y$;

(d') if $A \in \text{PN}_1$ and $tc(A) = X, \bar{a}, Y$ and $B \in \text{PN}_2$ and $tc(B) = Z, \overset{+}{b}$,
then $C \in \text{PN}_1$, where C results from applying \2 to $\overset{+}{b}$ and $\bar{a}$;
$tc(C) = X, Z, \bar{b}\backslash a, Y$;

(e') if $A \in \text{PN}_2$ and $tc(A) = X, \overset{+}{b}, Y$ and $B \in \text{PN}_1$ and $tc(B) = \bar{a}, Z$,
then $C \in \text{PN}_1$, where C results from applying \2 to $\overset{+}{b}$ and $\bar{a}$;
$tc(C) = X, \bar{b}\backslash a, Z, Y$;

(f') if $A \in \text{PN}_1$ and $tc(A) = X, \bar{a}, Y$ and $B \in \text{PN}_2$ and $tc(B) = \overset{+}{b}, Z$,
then $C \in \text{PN}_1$, where C results from applying /2 to $\bar{a}$ and $\overset{+}{b}$;
$tc(C) = X, a/\bar{b}, Z, Y$;

(g') if $A \in \text{PN}_2$ and $tc(A) = X, \overset{+}{b}, Y$ and $B \in \text{PN}_1$ and $tc(B) = Z, \bar{a}$,
then $C \in \text{PN}_1$, where C results from applying /2 to $\bar{a}$ and $\overset{+}{b}$;
$tc(C) = X, Z, a/\bar{b}, Y$;

(h') if $A \in \text{PN}_1$ and $tc(A) = X, \overset{+}{at}, Y$, then $C \in \text{PN}_2$, where C results from applying ax to $\overset{+}{at}$ and the non-connected occurrence of $\bar{at}$ in A;
$tc(C) = X, \overset{+}{at}, Y$.

Note (i) that for all $A \in \text{PN}_1$: $go(A)$ is an atomic category; (ii) that every $A \in \text{PN}_1$ contains exactly two occurrences of an atomic category at which are not connected by ax, viz., $go(A)$ and one negatively signed occurrence of at; and (iii) that if $C \in \text{PN}_2$, then all occurrences of atomic categories in C are connected by ax. Hence clause (h') yields a $C \in \text{PN}_2$ for every $A \in \text{PN}_1$.

Let $C \in \text{PN}_2$ be such that $go(C) = at$. The ax link which connects $go(C)$ with some negatively signed occurrence of at will be called the *goal link* of C, and C^- will denote the structure from which C is obtained by adding a goal link. Obviously, if $C \in \text{PN}_2$ is obtained from $A \in \text{PN}_1$ by clause (h'), then $A = C^-$.

Now, the set PN_1 is equal to $\{ C^- \mid C \in \text{PN} \,\&\, go(C) \in \text{ATOM} \}$, while PN_2 is the same set as PN. This is shown by induction on the degree of $tc(C)$:

• Observe that (b) through (g) and (b') through (g') yield structures C with $d(tc(C)) > 0$. Consequently, if $d(tc(C)) = 0$, then: (i) $C \in \text{PN}$ must be due to (a), therefore $C = \overline{at}\ \overline{at}$ or $C = \overline{at}\ \overline{at}$; and (ii) $C \in \text{PN}_2$ must be due to (h'), i.e., C is obtained from $C^- \in \text{PN}_1$, where $d(tc(C^-)) = d(tc(C)) = 0$. So, $C^- \in \text{PN}_1$ must be due to (a'): $C^- = \overline{at}\ \overline{at}$ (and $C = \overline{at}\ \overline{at}$) or $C^- = \overline{at}\ \overline{at}$ (and $C = \overline{at}\ \overline{at}$).

• If $d(tc(C)) > 0$, then either (i) $go(C) = b\backslash a$ (or, analogously, a/b); or (ii) $go(C) \in \text{ATOM}$.

◦ (i) If $go(C) = b\backslash a$ and $C \in \text{PN}$, then $tc(C) = X, \overset{+}{b\backslash a}, Y$, and – by an 'easily verifiable' fact[19] – for some $A \in \text{PN}$ with $tc(A) = X, \overset{+}{a}, \overline{b}, Y$: C can be obtained by applying (b) to A. Conversely, $C \in \text{PN}_2$ *must* be obtained by applying (b') to $A \in \text{PN}_2$. Since $A \in \text{PN}$ iff $A \in \text{PN}_2$ by induction hypothesis, we have that $C \in \text{PN}$ iff $C \in \text{PN}_2$.

◦ (ii) If $go(C) \in \text{ATOM}$, then $C \in \text{PN}$ is due to (d), (e), (f) or (g). Suppose (d) is involved. Then $tc(C) = X, Z, \overset{-}{b\backslash a}, Y$, and C is the result of applying $\backslash 2$ to $\overset{+}{b}$ in $B \in \text{PN}$ and $\overline{a}$ in $A \in \text{PN}$, where $tc(B) = Z, \overset{+}{b}$ and $tc(A) = X, \overline{a}, Y$. By induction hypothesis: $B \in \text{PN}_2$ and $A \in \text{PN}_2$. Since $go(B) = b$, we have that $at = go(A)$. Therefore, $A \in \text{PN}_2$ must have been obtained by (h') from $A^- \in \text{PN}_1$, i.e., A with missing goal link. By (d'), we can apply $\backslash 2$ to $\overline{a}$ in $A^- \in \text{PN}_1$ and $\overset{+}{b}$ in $B \in \text{PN}_2$, yielding $C^- \in \text{PN}_1$, i.e., C with missing goal link, which, by (h'), can be added so as to obtain C as a member of PN_2. (The cases (e), (f) and (g) are analogous.) Conversely, $C \in \text{PN}_2$ must

[19] Viz., the proof net version of observation **B** made in section 3 above: If $C \in \text{PN}$ and $tc(C) = X, \overset{+}{b\backslash a}, Y$ (or $X, \overset{+}{a/b}, Y$), then there is an $A \in \text{PN}$ with $tc(A) = X, \overset{+}{a}, \overline{b}, Y$ (or $X, \overline{b}, \overset{+}{a}, Y$) such that C results from applying (b) (or (c)) to A. (Roorda (1991), p. 33)

have been obtained by (h') from $C^- \in \text{PN}_1$. Note that $tc(C^-) = tc(C)$ and $d(tc(C)) > 0$, so $C^- \in \text{PN}_1$ must be due to (d'), (e'), (f') or (g'). If (d') is responsible, then $tc(C^-) = X, Z, \overset{-}{b}\backslash a, Y$, and C^- results from applying $\backslash 2$ to $\overset{+}{b}$ in $B \in \text{PN}_2$ and $\overset{-}{a}$ in $A' \in \text{PN}_1$, where $tc(A') = X, \overset{-}{a}, Y$ and $tc(B) = Z, \overset{+}{b}$. By induction hypothesis: $A' = A^-$ for $A \in \text{PN}$, while $B \in \text{PN}$. Therefore, by (d), we can apply $\backslash 2$ to $\overset{+}{b}$ in $B \in \text{PN}$ and $\overset{-}{a}$ in $A \in \text{PN}$, yielding C^- with additional goal link – i.e., C – as a member of PN. (The cases (e'), (f') and (g') are analogous.) □

It was said above that the definition of the set of proof nets in terms of (a') through (h') has two pleasant properties: (i) it allows a direct, inductive definition of the semantic interpretation of proof nets, and (ii) it brings out the correspondence between the application of rules in **L*** and the addition of links to proof nets.

As for (i): let u and v be different variables; let the sets of variables occurring in A and B be disjoint; let w be a variable which does not occur in A and B; and let $A[v{:=}\gamma]$ denote the structure which results from $A \in \text{PN}_1$ by replacing all occurrences of the variable v in A by the term γ. Then the semantic interpretation of proof nets can be defined as follows:

(a'') $\overset{-}{at}{:}u\ \overset{+}{at}{:}v \in \text{PN}_1$ and $\overset{+}{at}{:}v\ \overset{-}{at}{:}u \in \text{PN}_1$;
$tc(\overset{-}{at}{:}u\ \overset{+}{at}{:}v) = \overset{-}{at}{:}u, \overset{+}{at}{:}v$ and $tc(\overset{+}{at}{:}v\ \overset{-}{at}{:}u) = \overset{+}{at}{:}v, \overset{-}{at}{:}u$;

(b'') if $A \in \text{PN}_2$ and $tc(A) = X, \overset{+}{a}{:}\alpha, \overset{-}{b}{:}v, Y$ and X or Y is non-empty, then $C \in \text{PN}_2$, where C results from applying $\backslash 1$ to $\overset{+}{a}{:}\alpha$ and $\overset{-}{b}{:}v$ in A; $tc(C) = X, \overset{+}{b}\backslash a{:}\lambda v.\alpha, Y$;

(c'') if $A \in \text{PN}_2$ and $tc(A) = X, \overset{-}{b}{:}v, \overset{+}{a}{:}\alpha, Y$ and X or Y is non-empty, then $C \in \text{PN}_2$, where C results from applying $/1$ to $\overset{-}{b}{:}v$ and $\overset{+}{a}{:}\alpha$ in A; $tc(C) = X, a/\overset{+}{b}{:}\lambda v.\alpha, Y$;

(d'') if $A \in \text{PN}_1$ and $tc(A) = X, \overset{-}{a}{:}v, Y$ and $B \in \text{PN}_2$ and $tc(B) = Z, \overset{+}{b}{:}\beta$, then $C \in \text{PN}_1$, where C results from applying $\backslash 2$ to

$\overset{+}{b}{:}\beta$ in B and $\overset{-}{a}{:}w(\beta)$ in $A[v{:=}w(\beta)]$;

$tc(C) = X', Z, b\overset{-}{\backslash}a{:}w, Y'$, where $X', \overset{-}{a}{:}w(\beta), Y' = tc(A[v{:=}w(\beta)])$;

(e'') if $A \in \text{PN}_2$ and $tc(A) = X, \overset{+}{b}{:}\beta, Y$ and $B \in \text{PN}_1$ and $tc(B) = \overset{-}{a}{:}v, Z$,

then $C \in \text{PN}_1$, where C results from applying $\backslash 2$ to

$\overset{+}{b}{:}\beta$ in A and $\overset{-}{a}{:}w(\beta)$ in $B[v{:=}w(\beta)]$;

$tc(C) = X, b\overset{-}{\backslash}a{:}w, Z', Y$, where $\overset{-}{a}{:}w(\beta), Z' = tc(B[v{:=}w(\beta)])$;

(f'') if $A \in \text{PN}_1$ and $tc(A) = X, \overset{-}{a}{:}v, Y$ and $B \in \text{PN}_2$ and $tc(B) = \overset{+}{b}{:}\beta, Z$,

then $C \in \text{PN}_1$, where C results from applying $/2$ to

$\overset{-}{a}{:}w(\beta)$ in $A[v{:=}w(\beta)]$ and $\overset{+}{b}{:}\beta$ in B;

$tc(C) = X', a\overset{-}{/}b{:}w, Z, Y'$, where $X', \overset{-}{a}{:}w(\beta), Y' = tc(A[v{:=}w(\beta)])$;

(g'') if $A \in \text{PN}_2$ and $tc(A) = X, \overset{+}{b}{:}\beta, Y$ and $B \in \text{PN}_1$ and $tc(B) = Z, \overset{-}{a}{:}v$,

then $C \in \text{PN}_1$, where C results from applying $/2$ to

$\overset{-}{a}{:}w(\beta)$ in $B[v{:=}w(\beta)]$ and $\overset{+}{b}{:}\beta$ in A;

$tc(C) = X, Z', a\overset{-}{/}b{:}w, Y$, where $Z', \overset{-}{a}{:}w(\beta) = tc(B[v{:=}w(\beta)])$;

(h'') if $A \in \text{PN}_1$, $tc(A) = X, \overset{+}{at}{:}v, Y$ and $\overset{-}{at}{:}\gamma$ in A is not connected by ax,

then $C \in \text{PN}_2$, where C results from applying ax to

$\overset{+}{at}{:}\gamma$ and $\overset{-}{at}{:}\gamma$ in $A[v{:=}\gamma]$;

$tc(C) = tc(A[v{:=}\gamma])$.

The proof nets C defined in (b''), (c''), (d''), (e''), (f''), (g''), and (h'') will be called $\backslash 1(A)$, $/1(A)$, $\backslash 2(A^{w/v}, B)$, $\backslash 2(A, B^{w/v})$, $/2(A^{w/v}, B)$, $/2(A, B^{w/v})$ and $ax(A)$, respectively.

Though it is not exactly simple, the above definition is straightforward in comparison with the indirect semantic interpretation procedure given in Roorda (1991, pp. 34–38):

First, the set of *proof frames* PF is defined as the smallest set such that:

(A) if X is a list of signed atomic categories, then $X \in \text{PF}$; $tc(X) = X$;

(B) if $A \in \text{PF}$ and $tc(A) = X, \overset{+}{a}, \overset{-}{b}, Y$, then $C \in \text{PF}$, where C results from

applying \1 to $\overset{+}{a}$ and $\overset{-}{b}$ in A; $tc(C) = X, b\overset{+}{\backslash}a, Y$;

(C) if $A \in$ PF and $tc(A) = X, \overset{-}{b}, \overset{+}{a}, Y$, then $C \in$ PF, where C results from applying /1 to $\overset{-}{b}$ and $\overset{+}{a}$ in A; $tc(C) = X, a\overset{+}{/}b, Y$;

(D) if $A \in$ PF and $tc(A) = X, \overset{+}{b}, \overset{-}{a}, Y$, then $C \in$ PF, where C results from applying \2 to $\overset{+}{b}$ and $\overset{-}{a}$ in A; $tc(C) = X, b\overset{-}{\backslash}a, Y$;

(E) if $A \in$ PF and $tc(A) = X, \overset{-}{a}, \overset{+}{b}, Y$, then $C \in$ PF, where C results from applying /2 to $\overset{-}{a}$ and $\overset{+}{b}$ in A; $tc(C) = X, a\overset{-}{/}b, Y$.

Moreover, a *proof structure* is defined as a proof frame together with a linking of all its atomic categories by *ax* links, and a *planar* proof structure is defined as a proof structure of which the *ax* links do not cross.[20]

Second, for every category occurrence in a proof structure C a lambda term is constructed:

(I) every category occurrence in $tc(C)$ is assigned a different variable;

(II) the assignment of terms to the conclusions of \2, /2, \1 and /1 links in C is extended to the premises of these links in the following way:

$$\frac{\overset{+}{b}{:}?\quad \overset{-}{a}{:}¿}{b\overset{-}{\backslash}a{:}\gamma}\backslash 2 \qquad \frac{\overset{-}{a}{:}¿\quad \overset{+}{b}{:}?}{a\overset{-}{/}b{:}\gamma}/2$$

In \2 and /2 links, the category $\overset{+}{b}$ is assigned a fresh variable u, and the term assigned to $\overset{-}{a}$ is $\gamma(u)$.

$$\frac{\overset{+}{a}{:}¿\quad \overset{-}{b}{:}?}{b\overset{+}{\backslash}a{:}\delta}\backslash 1 \qquad \frac{\overset{-}{b}{:}?\quad \overset{+}{a}{:}¿}{a\overset{+}{/}b{:}\delta}/1$$

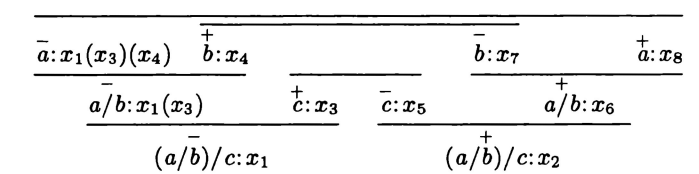
[20]Though a proof net also consists of linked signed categories of which the atomic ones are all connected by non-crossing *ax* links, the set of planar proof structures is considerably larger than the set PN defined earlier. This is due to the fact that on account of the clauses (B) through (E) above, the links \1, /1, \2 and /2 are allowed to connect any pair of adjacent (and appropriately signed) categories in a proof structure. The clauses (*b*) through (*g*), on the other hand, have the effect that the links \1 and /1 only connect adjacent *terminal* categories of a *single* proof net, whereas the links \2 and /2 only connect adjacent *terminal* categories of *separate* proof nets.

In \1 and /1 links, the respective terms assigned to $\bar{b}$ and $\overset{+}{a}$ are two different fresh variables u and v, and the conclusion term δ is *rewritten*[21] as $\lambda u.v$.

(III) every *ax* link in C which connects $\overset{+}{at}{:}\gamma$ and $\bar{at}{:}\delta$ induces a substitution[22] of γ by δ. This substitution is *global*: it involves all occurrences of γ in C.

Third, Roorda supplies a list of five[23] 'proof net conditions' which a proof structure interpreted in the way indicated in (I) through (III) must observe in order to be a proof net, and shows that the set of proof nets corresponding to **L**-derivable sequents is identical to the set of planar proof structures which satisfy these conditions. Except for the first one (which requires that there be exactly one terminal category signed '+'), these conditions constrain the

[21]Positively signed categories are always assigned variables, so this rewriting of δ is essentially a *substitution*. Moreover, though Roorda (1991) does not mention this, the substitution will have to be *global*: just as the substitution specified in (III) below, it must affect *all* occurrences of δ in the proof structure. This can be seen as follows:



Stage (I) yields the initial assignment of x_1 and x_2; the clause for /2 in stage (II) takes care of the assignment of $x_1(x_3)$, x_3, $x_1(x_3)(x_4)$ and x_4. Now, the clause for /1 assigns x_5 and x_6, substitutes x_2 by $\lambda x_5.x_6$, assigns x_7 and x_8, and substitutes x_6 by $\lambda x_7.x_8$. If the latter substitution is local and does not affect the occurrence of x_6 in $\lambda x_5.x_6$, then after (III) (which – globally – replaces x_8, x_3 and x_4 by $x_1(x_3)(x_4)$, x_5, and x_7, respectively) the undesirable term $\lambda x_5.x_6$ is obtained for the goal of the proof net. Global substitution, on the other hand, will eventually yield the right result, viz., $\lambda x_5 \lambda x_7.x_1(x_5)(x_7)$.

[22]Roorda says that γ and δ are *unified* but that γ is always a variable, so 'the unification is nothing else than the substitution [of γ by δ]' (1991, p. 35). The notion of unification is probably invoked for the reason that Roorda wants the process to fail, 'typically' when γ occurs in δ. 'In that case we draw the conclusion that the proof structure is not a proof net, and we do not provide a lambda term.' (1991, p. 35). (Note that a substitution under such circumstances would be impeccable.) However, the fact that this eventuality does not arise in $\mathrm{PN}_1 \cup \mathrm{PN}_2$ kept us from taking over this subtlety.

[23]In fact, a sixth proof net condition enforces the 'non-empty antecedent property' alluded to in footnote 18 above.

assignment of lambda terms to proof structures.

Their precise contents need not concern us here, since the equivalence of the interpretation procedures specified in (a'') through (h'') and (I) through (III) for members C of the set $\mathrm{PN}_1 \cup \mathrm{PN}_2$ can be established directly:[24] it is easy to see that both (I) through (III) and (a'') through (h'') license an assignment of lambda terms to category occurrences in $C \in \mathrm{PN}_1 \cup \mathrm{PN}_2$ iff:

- every $\bar{c}$ occurring in $tc(C)$ is assigned a different variable;
- the assignment to every $\backslash 1$, $/1$, $\backslash 1$ and $/2$ link in C is as follows:

$$\frac{\overset{+}{b}:\beta \quad \overset{-}{a}:\alpha(\beta)}{\overset{-}{b\backslash a}:\alpha}\backslash 2 \qquad \frac{\overset{-}{a}:\alpha(\beta) \quad \overset{+}{b}:\beta}{\overset{-}{a/b}:\alpha}/2$$

$$\frac{\overset{+}{a}:\gamma \quad \overset{-}{b}:v}{\overset{+}{b\backslash a}:\lambda v.\gamma}\backslash 1 \qquad \frac{\overset{-}{b}:v \quad \overset{+}{a}:\gamma}{\overset{+}{a/b}:\lambda v.\gamma}/1;\ \text{and}$$

- if an ax link connects $\overset{-}{at}:\gamma$ and $\overset{+}{at}:\delta$, then γ is identical to δ.

Do proof nets consitute a solution to the problem of spurious ambiguity? Observe that the proof net semantics defined above in (b'') through (h'') is closely related to **L***, viz., via the function k from **L*** proofs to the power set of $\mathrm{PN}_1 \cup \mathrm{PN}_2$ which is defined below.

As before, we let $\pi[T{:}T' \vdash c{:}\gamma]$ represent that π is an **L*** proof with interpreted conclusion sequent $T{:}T' \vdash c{:}\gamma$.

- $k(at^*{:}v \vdash at{:}v) = \{\overset{-}{at}{:}v \quad \overset{+}{at}{:}u \mid u \neq v\} \cup \{\overset{+}{at}{:}u \quad \overset{-}{at}{:}v \mid u \neq v\}$;
- for $\pi[b{:}v, T{:}T' \vdash a^*{:}\alpha]$: $k(\dfrac{\pi}{T{:}T' \vdash b\backslash a^*{:}\lambda v.\alpha}[\backslash R]) = \{\backslash 1(A) \mid A \in k(\pi)\}$;
- for $\pi[T{:}T', b{:}v \vdash a^*{:}\alpha]$: $k(\dfrac{\pi}{T{:}T' \vdash a/b^*{:}\lambda v.\alpha}[/R]) = \{/1(A) \mid A \in k(\pi)\}$;
- for $\pi_1[T{:}T' \vdash b^*{:}\beta]$ and $\pi_2[U{:}U', a^*{:}v, V{:}V' \vdash at{:}\gamma]$:

[24] Strictly speaking, the set PN_1 transcends the range of the clauses (I) through (III), because its members fail to be proof structures. However, they can be incorporated easily, for instance by allowing (I) through (III) to operate on members of the set PO (of *proof objects*): the smallest set which includes the set PF defined above and is closed under the operation of adding ax links.

$$k(\frac{\pi_1 \qquad\qquad \pi_2}{U{:}U', T{:}T', b\backslash a^*{:}w, V{:}V' \vdash at{:}\gamma[v{:=}w(\beta)]}\,[\backslash L]) =$$

$$\{\backslash 2(A^{w/v}, B) \mid A \in k(\pi_2)\ \&\ B \in k(\pi_1)\ \&\ tc(B) = \bar{T}{:}T', \overset{+}{b}{:}\beta\} \cup$$

$$\{\backslash 2(A, B^{w/v}) \mid A \in k(\pi_1)\ \&\ B \in k(\pi_2)\ \&\ tc(B) = \bar{a}{:}v, \bar{V}{:}V', \overset{+}{at}{:}u, \bar{U}{:}U'\};$$

$$k(\frac{\pi_1 \qquad\qquad \pi_2}{U{:}U', a/b^*{:}w, T{:}T', V{:}V' \vdash at{:}\gamma[v{:=}w(\beta)]}\,[/L]) =$$

$$\{/2(A^{w/v}, B) \mid A \in k(\pi_2)\ \&\ B \in k(\pi_1)\ \&\ tc(B) = \overset{+}{b}{:}\beta, \bar{T}{:}T'\} \cup$$

$$\{/2(A, B^{w/v}) \mid A \in k(\pi_1)\ \&\ B \in k(\pi_2)\ \&\ tc(B) = \bar{V}{:}V', \overset{+}{at}{:}u, \bar{U}{:}U', \bar{a}{:}v\};$$

- for $\pi[U{:}U', a^*{:}v, V{:}V' \vdash at{:}\gamma]$: $k(\frac{\pi}{U{:}U', a{:}v, V{:}V' \vdash at^*{:}\gamma}\,[*]) =$ $\{ax(A) \mid A \in k(\pi)\}$.

Next, for (non-asterisked) interpreted sequents $T{:}T' \vdash c{:}\gamma$, we define the set $cp(T{:}T' \vdash c{:}\gamma)$ of *cyclic permutations* of $T{:}T' \vdash c{:}\gamma$, as follows:

$$cp(T{:}T' \vdash c{:}\gamma) = \{\bar{T}_2{:}T'_2, \overset{+}{c}{:}\gamma, \bar{T}_1{:}T'_1 \mid T{:}T' = T_1{:}T'_1, T_2{:}T'_2\}.$$

Note that a proof net C corresponds to a sequent $T : T' \vdash c : \gamma$ if and only if $tc(C) \in cp(T : T' \vdash c : \gamma)$. By a straightforward induction on the length of π (which will be omitted here) it can be proven that for **L*** proofs $\pi[T{:}T' \vdash c^*{:}\gamma]$ and $\rho[U{:}U, c^*{:}v, V{:}V' \vdash at{:}\gamma]$:

$$k(\pi) = \{C \in \text{PN}_2 \mid tc(C) \in cp(T{:}T' \vdash c{:}\gamma)\};\ \text{and}$$

$$k(\rho) = \{A \in \text{PN}_1 \mid \text{for some } u{:}\ tc(A) \in cp(U{:}U', c{:}v, V{:}V' \vdash at{:}u) \text{ and } \bar{at}{:}\gamma \text{ occurs not connected by } ax \text{ in } A\}.$$

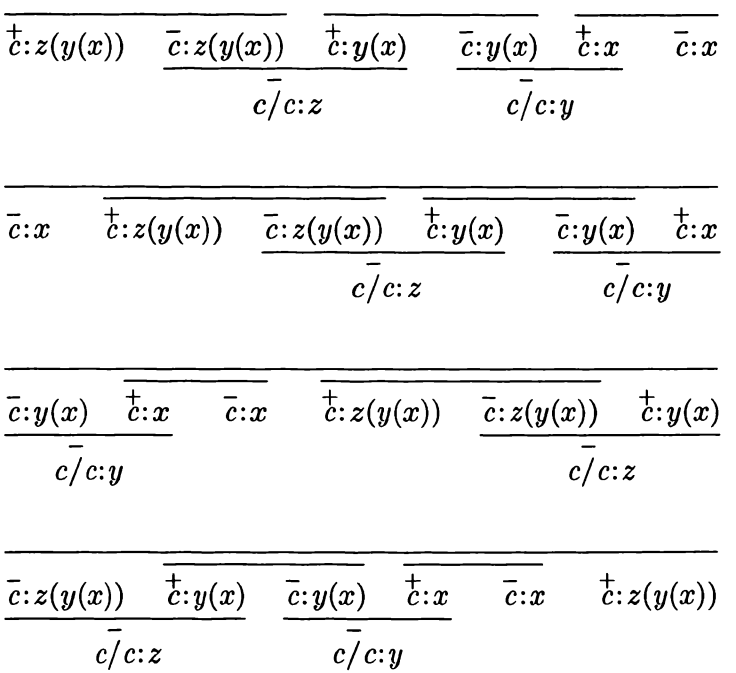
So, the function k assigns a subset of PN_2 to every **L*** proof of a sequent with an asterisked goal and a subset of PN_1 to every **L*** proof of a sequent with a non-asterisked goal.

Moreover, the set $\{k(\pi) \mid \pi \text{ is an } \mathbf{L^*} \text{ proof}\}$ partitions $\text{PN}_1 \cup \text{PN}_2$, for (*i*) it is obvious that every $C \in \text{PN}_1 \cup \text{PN}_2$ is a member of at least one $k(\pi)$; and (*ii*) Theorem 2 (see section 4 above) entails that every $C \in \text{PN}_1 \cup \text{PN}_2$ is a member of at most one $k(\pi)$.

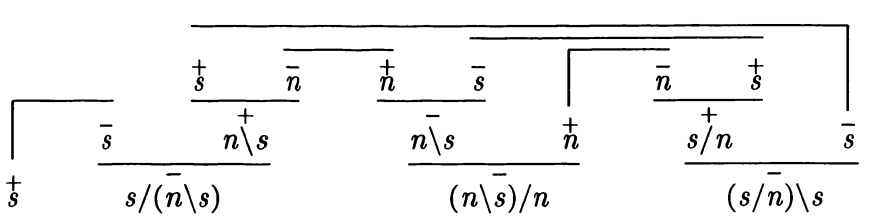
Now, since for every **L**-derivable sequent $c_1:v_1, \ldots, c_n:v_n \vdash c:\gamma$ there is one **L*** derivation of $c_1:v_1, \ldots, c_n:v_n \vdash c^*:\gamma'$ such that $\gamma = \gamma'$ and since for a sequent $c_1:v_1, \ldots, c_n:v_n \vdash c:\gamma$ the set $cp(c_1:v_1, \ldots, c_n:v_n \vdash c:\gamma)$ contains $n+1$ members, the above means that every reading of an **L**-derivable sequent $c_1, \ldots, c_n \vdash c$ corresponds to $n+1$ proof nets.

Accordingly, there are four proof nets for the sequent $c/c, c/c, c \vdash c$, and eight proof nets for $s/(n\backslash s), (n\backslash s)/n, (s/n)\backslash s \vdash s$.[25] Note that the proof nets in (47) make up $k(31)$, and that the proof nets in (48) and (49) constitute $k(32)$ and $k(33)$, respectively.

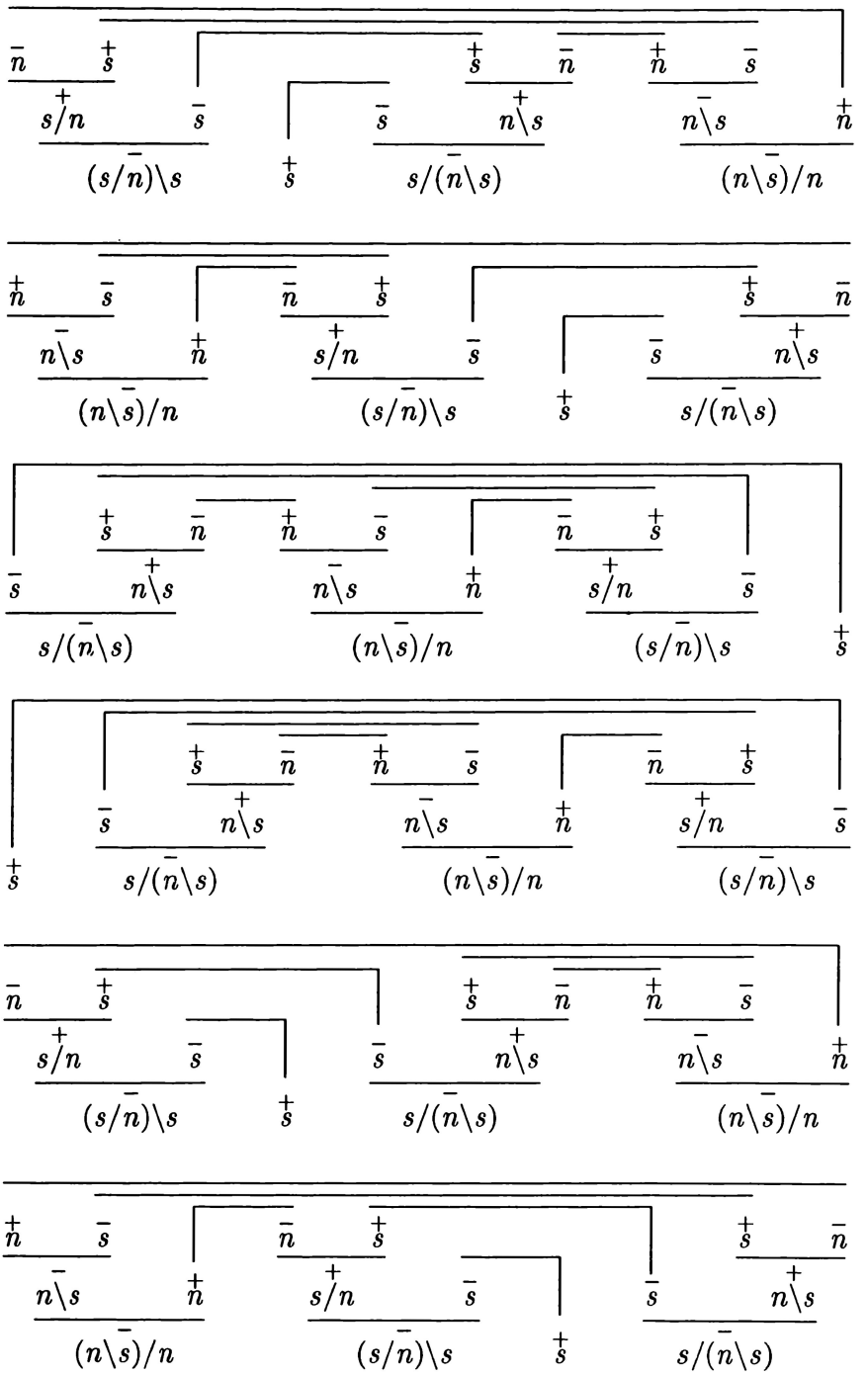
(47)

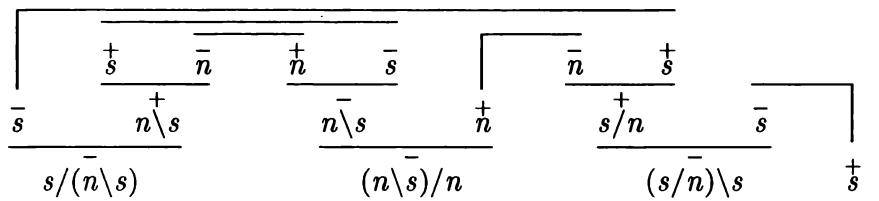


(48)



[25] As above (cf. footnote 10), we identify all proof nets which differ only with respect to the particular variables assigned.


(49)



If $s/(\overset{-}{n}\backslash s)$, $(\overset{-}{n}\backslash s)/n$, and $(\overset{-}{s}/n)\backslash s$ are assigned the variables u, v and w, then the term $u(\lambda u'.w(\lambda w'.v(u')(w')))$ is assigned to the positively signed terminal category s in the proof nets listed in (48), and $w(\lambda w'.u(\lambda u'.v(u')(w')))$ is assigned to the positively signed terminal category s in the proof nets listed in (49).

Roorda (1991, p. 27) claims that

> a proof net can be considered as a parallellized sequent proof and as such it lacks the spurious ambiguity of (even Cut-free) sequent proofs. A proof net is a concrete structure, not merely an abstract equivalence class of derivations [...].

However, we have just seen that the set of proof nets has to be partitioned into equivalence classes if it is to serve as a solution to the spurious ambiguity problem: for every reading of an **L**-derivable sequent $c_1, \ldots, c_n \vdash c$, there are $n + 1$ proof nets.

Of course, these $n + 1$ proof nets are equivalent in that they make up a set of cyclic permutations. But on the one hand, one cannot isolate one specific representant – e.g., the proof net with its positive terminal category in rightmost position – from this set, since one has to be strict on order and adjacency of premises in proof nets, so that all these permutations are required for the definition (cf. clause (*b*) and (*c*)). And on the other hand, it is not possible to consider this set of permutations as a monolithic object, since sometimes a specific representant (e.g., the proof net with its positive terminal category in rightmost position – cf. B in clause (*d*)) has to be isolated from this set.

So, ironically, it is the proof net approach which has to resort to 'abstract' equivalence classes, whereas sequent normalization (as well as partial deduction) associates one concrete object with each interpretation of a derivable sequent: its **L* (PD)** proof.

6 Encoding Atomic Categories

Ponse (1988) proved that **LP** derivability in an atomic goal category can be mimicked by **LP** derivability using one atomic category only. **LP** refers to the Lambek calculus with Permutation, which is also known as Lambek-Van Benthem calculus.[26] In the present section we will show that (a generalization of) this result can be extended to **L**, the Lambek calculus proper: **L** derivability in *any* category can be mimicked by **L** derivability using one atomic category only. More formally:

Let the set AT consist of the distinct atomic categories $at_1, \ldots, at_k$, let at be an atomic category, and let CAT_{AT} and $\text{CAT}_{\{at\}}$ be the sets of categories based on AT and $\{at\}$, respectively. Then there is a substitution σ replacing every $at_i \in \text{AT}$ by a $c_i \in \text{CAT}_{\{at\}}$ such that for all $c_1, \ldots, c_n, c$ in CAT_{AT}:

(50) $c_1, \ldots, c_n \vdash_{\mathbf{L}} c$ if and only if $\sigma(c_1, \ldots, c_n) \vdash_{\mathbf{L}} \sigma(c)$.

If σ is a substitution and α is a category or a sequence of categories, then the expression $\sigma(\alpha)$ denotes the result of performing σ to α. The atomic category at will be abbreviated as t in the sequel.

The proof of theorem (50) is organized as follows. First, the calculus **L*** will be used for establishing the facts (51) and (53), which express useful properties of **L**-derivable sequents that will be exploited throughout. Next, a Lemma will be proven which concerns the non-derivability of certain sequents that involve categories built up from the categories $(t/t)/t$, $((t/t)/(t/t))/(t/t)$ and atomic categories different from t. This Lemma is then shown to entail Claim 1, which states that the categories $(t/t)/t$ and $((t/t)/(t/t))/(t/t)$ can be used to encode two atomic categories, viz., t and some other atomic category, also in the presence of yet other atomic categories. Finally, the substitution σ employed in Claim 1 is generalized in Claim 2: by means of a substitution $\sigma_{\langle t, at_1, \ldots, at_m \rangle}$, any finite number of atomic categories $t, at_1, \ldots, at_m$ can be encoded in terms of t.[27]

[26] See Section 3.2 of Chapter 1, and Van Benthem (1988; 1991).

[27] The above theorem follows from Claim 2, since the following substitutions will meet the requirement specified in (50):

- $\sigma_{\langle t, at_1, \ldots, at_k \rangle}$ if $t \notin \text{AT}$ (note that any category based on AT is based on $\text{AT} \cup \{t\}$);
- $\sigma_{\langle t, at_1, \ldots, at_{i-1}, at_{i+1}, \ldots, at_k \rangle}$ if $t \in \text{AT}$ and $\text{AT} = \{at_1, \ldots, at_{i-1}, t, at_{i+1}, \ldots, at_k\}$.

We start by putting each category c in CAT_{AT} into an equivalence class $\lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$. Let c and $c_1, \ldots, c_{p+q}$ be members of CAT_{AT} (where $p+q \geq 0$), and let $at \in \text{AT}$. Then:

$c \in \lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$ iff

(*a*) $c = at$ and $p+q = 0$;
(*b*) $c = c_p\backslash c'$ and $c' \in \lfloor c_{p-1}\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$; or
(*c*) $c = c'/c_{p+q}$ and $c' \in \lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q-1}\rfloor$.

The sets $\lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$ partition CAT_{AT}.[28] Note that the following claims hold:

(51) If $c \in \lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$, then
$T \vdash_{\mathbf{L}} c$ iff $c_1, \ldots, c_p, T, c_{p+q}, \ldots, c_{p+1} \vdash_{\mathbf{L}} at$.

(52) $U, c^*, V \vdash_{\mathbf{L}^*} at$ and $c \in \lfloor c_p\backslash ...\backslash c_1\backslash at'/c_{p+1}/.../c_{p+q}\rfloor$ iff
$at' = at$; $U = T_1, \ldots, T_p$; $V = T_{p+q}, \ldots, T_{p+1}$;
and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}^*} c_i^*$.

Both (51) and (52) are proven by induction on $p+q$.

As for (51):
• If $p+q = 0$, then the claim is trivial.
• If $p+q > 0$, then (*i*) $c = c_p\backslash c'$ and $c' \in \lfloor c_{p-1}\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$; or (*ii*) $c = c'/c_{p+q}$ and $c' \in \lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q-1}\rfloor$. We only treat (*i*), since (*ii*) is analogous. Note that the following claims are equivalent: (1) $T \vdash_{\mathbf{L}} c_p\backslash c'$; (2) $T \vdash_{\mathbf{L}^*} c_p\backslash c'^*$; (3) $c_p, T \vdash_{\mathbf{L}^*} c'^*$; (4) $c_p, T \vdash_{\mathbf{L}} c'$; (5) $c_1, \ldots, c_p, T, c_{p+q}, \ldots, c_{p+1} \vdash_{\mathbf{L}} at$. Theorem 1.1 (see section 3 above) yields the equivalence of (1) and (2) as well as (3) and (4); the equivalence of (2) and (3) is due to $\mathbf{L}^*$; and (4) and (5) are equivalent on account of the induction hypothesis. □

As for (52):

[28] Different categories c and c' are members of the same set $\lfloor c_p\backslash ...\backslash c_1\backslash at/c_{p+1}/.../c_{p+q}\rfloor$ iff c and c' have the same final atomic value (viz., at) and the same series of left-hand side ($c_p, \ldots, c_1$) and right-hand side ($c_{p+1}, \ldots, c_{p+q}$) arguments, but combine with these arguments in a different order. The set $\lfloor t\backslash t/(t/t)/t\rfloor$, for example, consists of three categories: (a) $((t\backslash t)/(t/t))/t$, (b) $(t\backslash(t/(t/t)))/t$ and (c) $t\backslash((t/(t/t))/t)$.

• $p+q=0$. Then $U, c^*, V \vdash_{\mathbf{L}^*} at$ must be an axiom $at^* \vdash_{\mathbf{L}^*} at$; so $at' = at$ and U and V are empty.
• $p+q>0$. Then (*i*) $c = c_p \backslash c'$ and $c' \in \lfloor c_{p-1} \backslash ... \backslash c_1 \backslash at'/c_{p+1}/.../c_{p+q} \rfloor$; or (*ii*) $c = c'/c_{p+q}$ and $c' \in \lfloor c_p \backslash ... \backslash c_1 \backslash at'/c_{p+1}/.../c_{p+q-1} \rfloor$. We only treat (*ii*), since (*i*) is analogous. The sequent $U, c'/c_{p+q}{}^*, V \vdash at$ must be derived by $/L$ in $\mathbf{L}^*$. Hence $U, c'/c_{p+q}{}^*, V \vdash_{\mathbf{L}^*} at$ iff $U, c'^*, V' \vdash_{\mathbf{L}^*} at$ and $T_{p+q} \vdash_{\mathbf{L}^*} c_{p+q}{}^*$, where $V = T_{p+q}, V'$. By induction hypothesis: $U, c'^*, V' \vdash_{\mathbf{L}^*} at$ iff $at' = at$; $U = T_1, \ldots T_p$; $V' = T_{p+q-1}, \ldots T_{p+1}$; and for all i such that $1 \leq i \leq p+q-1$: $T_i \vdash_{\mathbf{L}^*} c_i{}^*$. □

Given (52), suppose that $T \vdash_{\mathbf{L}} at$. This holds iff $T \vdash_{\mathbf{L}^*} at^*$ by Theorem 1.1 (see Section 3 above). The sequent $T \vdash at^*$ must have been derived by the * rule in $\mathbf{L}^*$. Therefore, $T = U, c, V$ and $U, c^*, V \vdash_{\mathbf{L}^*} at$. Now, for some $c_1, \ldots, c_{p+q}, at'$: $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash at'/c_{p+1}/.../c_{p+q} \rfloor$. By (52), we have that $at' = at$; $U = T_1, \ldots, T_p$; $V = T_{p+q}, \ldots, T_{p+1}$; and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}^*} c_i{}^*$, which is equivalent to $T_i \vdash_{\mathbf{L}} c_i$ by Theorem 1.1. Summing up:

(53) $T \vdash_{\mathbf{L}} at$ iff there is a $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash at/c_{p+1}/.../c_{p+q} \rfloor$ such that $T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$ and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$.

Let A be a set of categories. In the sequel we will say that c is an A-category iff c is built up from categories in A.[29] We will abbreviate sequences $t/t, \ldots, t/t$ consisting of n occurrences of the category t/t as $(t/t)^n$.

Lemma: let A be the set $\{at_1, \ldots, at_k, (t/t)/t, ((t/t)/(t/t))/(t/t)\}$, where $at_1, \ldots, at_k$ and t are distinct atomic categories; and let T be a non-empty sequence of A-categories. Then:

(**a**) $T, t \not\vdash_{\mathbf{L}} t$;
(**b**) for all $n \in \mathbb{N}$: $T, (t/t)^n \not\vdash_{\mathbf{L}} t$; and
(**c**) $T, t/t, t \not\vdash_{\mathbf{L}} t$.

Proof of (**a**) and (**b**): by induction on m, the number of occurrences of $at_1, \ldots, at_k$, $(t/t)/t$ and $((t/t)/(t/t))/(t/t)$ in T.

• $m=1$. Then $T = at_i$ $(1 \leq i \leq k)$; $T = (t/t)/t$; or $T = ((t/t)/(t/t))/(t/t)$:

[29] That is, the set of A-categories is the smallest set A' such that (*i*) $A \subseteq A'$; and (*ii*) if $c \in A'$ and $c' \in A'$, then $c/c' \in A'$ and $c' \backslash c \in A'$.

(**a**) $at_i, t \not\vdash_{\mathbf{L}} t$; $(t/t)/t, t \not\vdash_{\mathbf{L}} t$; and $((t/t)/(t/t))/(t/t), t \not\vdash_{\mathbf{L}} t$.

(**b**) That $T, (t/t)^n \not\vdash_{\mathbf{L}} t$ can be shown by *at*-count, a notion introduced in Van Benthem (1986). Let $at \in$ AT. Then *at*-count$[c]$ is defined as follows: for $c \in$ AT, *at*-count$[c] = 1$ if $c = at$, while *at*-count$[c] = 0$ if $c \neq at$; *at*-count$[a/b]$ $=$ *at*-count$[b\backslash a]$ $=$ *at*-count$[a]$ $-$ *at*-count$[b]$. Moreover, *at*-count$[c_1, \ldots, c_n]$ $=$ *at*-count$[c_1] + \ldots +$ *at*-count$[c_n]$. A useful property of **L**-derivable sequents $T \vdash c$ is that for all $at \in$ AT: *at*-count$[T]$ $=$ *at*-count$[c]$. (The proof proceeds by an easy induction on the length of the derivation of $T \vdash c$.)

Note that t-count$[(t/t)/t] = -1$; that t-count$[((t/t)/(t/t))/(t/t)] = 0$; and that for all $i \in \{1, \ldots, k\}$ and $n \in \mathbb{N}$: t-count$[at_i] = t$-count$[(t/t)^n] = 0$. Hence t-count$[(t/t)/t, (t/t)^n] = -1$ and t-count$[((t/t)/(t/t))/(t/t), (t/t)^n] =$ t-count$[at_i, (t/t)^n] = 0$. On the other hand, t-count$[t] = 1$. So, for all $n \in \mathbb{N}$: $at_i, (t/t)^n \not\vdash_{\mathbf{L}} t$; $(t/t)/t, (t/t)^n \not\vdash_{\mathbf{L}} t$; and $((t/t)/(t/t))/(t/t), (t/t)^n \not\vdash_{\mathbf{L}} t$.

• $m > 1$. Note[30] that if c is an A-category and $c \in \lfloor c_p\backslash \ldots \backslash c_1 \backslash t/c_{p+1}/\ldots/c_{p+q} \rfloor$, then (*i*) $c_{p+1} = c_{p+2} = t$, so $c \in \lfloor c_p\backslash \ldots \backslash c_1\backslash t/t/t/c_{p+3}/\ldots/c_{p+q} \rfloor$; or (*ii*) c_{p+1} $= t$ and $c_{p+2} = c_{p+3} = t/t$, so $c \in \lfloor c_p\backslash \ldots \backslash c_1\backslash t/t/(t/t)/(t/t)/c_{p+4}/\ldots/c_{p+q} \rfloor$.

(**a**) Suppose $T, t \vdash_{\mathbf{L}} t$. By (53), there is a $c \in \lfloor c_p\backslash \ldots \backslash c_1\backslash t/c_{p+1}/\ldots/c_{p+q} \rfloor$ such that $T, t = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$ and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$. Since T is non-empty, this c cannot be the rightmost category t in T, t. Hence c is an A-category in T and either (*i*) $c \in \lfloor c_p\backslash \ldots \backslash c_1\backslash t/t/t/c_{p+3}/\ldots/c_{p+q} \rfloor$; or (*ii*) $c \in \lfloor c_p\backslash \ldots \backslash c_1\backslash t/t/(t/t)/(t/t)/c_{p+4}/\ldots/c_{p+q} \rfloor$. Focus on T_{p+2}. On the one hand: if (*i*), then $c_{p+2} = t$, so $T_{p+2} \vdash_{\mathbf{L}} t$; and if (*ii*), then $c_{p+2} = t/t$, so $T_{p+2} \vdash_{\mathbf{L}} t/t$. On the other hand: T_{p+2} is non-empty, since $T_{p+2} \vdash_{\mathbf{L}} c_{p+2}$; T_{p+2} is a sequence of A-categories, since t in T, t is part of T_{p+1} (which must be non-empty since $T_{p+1} \vdash_{\mathbf{L}} c_{p+1}$); and T_{p+2} contains less occurrences of $at_1, \ldots, at_k$, $((t/t)/(t/t))/(t/t)$ and $(t/t)/t$ than T, since c occurs in T but not in T_{p+2}. Therefore, the induction hypothesis for (**b**) ($n = 0$) yields that $T_{p+2} \not\vdash_{\mathbf{L}} t$, while the induction hypothesis for (**a**) yields that $T_{p+2}, t \not\vdash_{\mathbf{L}} t$. Because $t/t \in \lfloor t/t \rfloor$, the latter entails – by (51) – that $T_{p+2} \not\vdash_{\mathbf{L}} t/t$. So, both (*i*) and (*ii*) lead to contradiction, which means that $T, t \not\vdash_{\mathbf{L}} t$.

(**b**) Suppose $T, (t/t)^n \vdash_{\mathbf{L}} t$. By (53), there is a $c \in \lfloor c_p\backslash \ldots \backslash c_1\backslash t/c_{p+1}/\ldots/c_{p+q} \rfloor$ such that $T, (t/t)^n = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$ and for all i, $1 \leq i \leq p+q$:

[30] This is easily seen by induction on the number of occurrences of $at_1, \ldots, at_k$, $(t/t)/t$ and $((t/t)/(t/t))/(t/t)$ in c.

$T_i \vdash_{\mathbf{L}} c_i$. Since T is non-empty, this c cannot be a category in $(t/t)^n$. Hence c is an A-category in T and either (i) $c \in \lfloor c_p\backslash ...\backslash c_1\backslash t/t/t/c_{p+3}/.../c_{p+q}\rfloor$; or (ii) $c \in \lfloor c_p\backslash ...\backslash c_1\backslash t/t/(t/t)/(t/t)/c_{p+4}/.../c_{p+q}\rfloor$. Focus on T_{p+1}. On the one hand: both (i) and (ii) entail that $c_{p+1} = t$, so $T_{p+1} \vdash_{\mathbf{L}} t$. On the other hand: T_{p+1} cannot be of the form $(t/t)^m$ for $m \leq n$, since $(t/t)^m$ and t have different t-counts; hence T_{p+1} consists of a non-empty subsequence T' of T followed by $(t/t)^n$, where T' contains less occurrences of at_1, ..., at_k, $((t/t)/(t/t))/(t/t)$ and $(t/t)/t$ than T, since c occurs in T but not in T_{p+1}. Therefore, the induction hypothesis of (**b**) yields that $T', (t/t)^n \nvdash_{\mathbf{L}} t$ in both cases. Since $T', (t/t)^n = T_{p+1}$, we have a contradiction. Consequently, $T, (t/t)^n \nvdash_{\mathbf{L}} t$. □

Proof of (**c**):
Suppose $T, t/t, t \vdash_{\mathbf{L}} t$. By (53), there is a $c \in \lfloor c_p\backslash ...\backslash c_1\backslash t/c_{p+1}/.../c_{p+q}\rfloor$ such that $T, t/t, t = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$ and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$. Since T is non-empty, this c cannot be t/t or t in $T, t/t, t$. Hence c is an A-category in T and either (i) $c \in \lfloor c_p\backslash ...\backslash c_1\backslash t/t/t/c_{p+3}/.../c_{p+q}\rfloor$; or (ii) $c \in \lfloor c_p\backslash ...\backslash c_1\backslash t/t/(t/t)/(t/t)/c_{p+4}/.../c_{p+q}\rfloor$.

Suppose (i). Then on the one hand: $c_{p+2} = t$, so $T_{p+2} \vdash_{\mathbf{L}} t$. But on the other hand: $T_{p+1} \vdash_{\mathbf{L}} t$ entails that T_{p+1} is non-empty and includes at least t. Hence t/t must be part of (a) T_{p+1} or (b) T_{p+2}. Suppose (a). Then T_{p+2} is a sequence of A-categories which is, moreover, non-empty since $T_{p+2} \vdash_{\mathbf{L}} c_{p+2}$, so that $T_{p+2} \vdash_{\mathbf{L}} t$ contradicts Lemma (**b**) ($n = 0$). Suppose (b). Then T_{p+2} consists of a sequence T' of A-categories followed by t/t and T' must be non-empty since $t/t \nvdash_{\mathbf{L}} t$, so that $T_{p+2} \vdash_{\mathbf{L}} t$ contradicts Lemma (**b**) ($n = 1$).

Suppose (ii). Then on the one hand: $c_{p+3} = t/t$, so $T_{p+3} \vdash_{\mathbf{L}} t/t$ and $T_{p+3}, t \vdash_{\mathbf{L}} t$ by (51). But on the other hand: $T_{p+1} \vdash_{\mathbf{L}} t$ entails that T_{p+1} is non-empty and includes at least t. Hence t/t must be part of T_{p+1} or T_{p+2}. Anyway, T_{p+3} is a sequence of A-categories which is, moreover, non-empty since $T_{p+3} \vdash_{\mathbf{L}} c_{p+3}$, so that $T_{p+3}, t \vdash_{\mathbf{L}} t$ contradicts Lemma (**a**).

All cases lead to contradiction, so $T, t/t, t \nvdash_{\mathbf{L}} t$. □

Corollary:
(1) There is no sequence S of A-categories such that
$S, t, t = T''', T'', T'$, where $T''' \vdash_{\mathbf{L}} t/t$, $T'' \vdash_{\mathbf{L}} t/t$ and $T' \vdash_{\mathbf{L}} t$.
Suppose the contrary. Then T''', T'' and T' are non-empty, so the second t in S, t, t is part of T', and the first t is part of T'' or T'. Either way T''' is a non-empty sequence of A-categories. But $T''' \vdash_{\mathbf{L}} t/t$ entails $T''', t \vdash_{\mathbf{L}} t$ by (51), and the latter contradicts Lemma (**a**).

(2) There is no sequence S of A-categories such that
$S, t/t, t/t, t = T'', T'$, where $T'' \vdash_{\mathbf{L}} t$ and $T' \vdash_{\mathbf{L}} t$.

Suppose the contrary. Then T'' and T' are non-empty, so the category t in $S, t/t, t/t, t$ is part of T', so that $T'' = S', (t/t)^m$, where $m \in \{0, 1, 2\}$ and S' is (a subsequence of) S. But then $T'' \not\vdash_{\mathbf{L}} t$, since $(t/t)^m \not\vdash_{\mathbf{L}} t$ by t-count, and for non-empty S': $S', (t/t)^m \not\vdash_{\mathbf{L}} t$ by Lemma (**b**).

(3) There is no non-empty sequence S of A-categories such that
$S, t, t = T'', T'$, where $T'' \vdash_{\mathbf{L}} t$ and $T' \vdash_{\mathbf{L}} t$.

Suppose the contrary. Then T'' and T' are non-empty, so the second t in S, t, t is part of T', and (*i*) $T'' = S, t$; or (*ii*) $T'' = S'$ and non-empty S' is (a subsequence of) S. Now, (*ii*) contradicts Lemma (**b**) ($n = 0$), and (*i*) contradicts Lemma (**a**) for non-empty S. Hence S is empty (and $T'' = T' = t$).

(4) There is no non-empty sequence S of A-categories such that
$S, t/t, t/t, t = T''', T'', T'$, where $T''' \vdash_{\mathbf{L}} t/t$, $T'' \vdash_{\mathbf{L}} t/t$ and $T' \vdash_{\mathbf{L}} t$.

Suppose the contrary. Then T''', T'' and T' are non-empty, and T''' is not a subsequence of S, since $T''' \vdash_{\mathbf{L}} t/t$ entails that $T''', t \vdash_{\mathbf{L}} t$ by (51), contradicting Lemma (**a**). So T''' includes the first t/t in $S, t/t, t/t, t$ – but not the second one, for then T'' or T' would have to be empty. Hence $T''' = S, t/t$ (so that $T'' = t/t$ and $T' = t$) and S is empty, since $S, t/t \vdash_{\mathbf{L}} t/t$ entails $S, t/t, t \vdash_{\mathbf{L}} t$ by (53), and the latter is impossible for non-empty S on account of Lemma (**c**).

Let t and at_0 be two distict atomic categories. The following claim shows that the compound categories $(t/t)/t$ and $((t/t)/(t/t))/(t/t)$ can be used for encoding t and at_0, respectively.

Claim 1:
Let the set $\text{AT} = \{t, at_0, at_1, \ldots, at_k\}$ consist of distinct atomic categories; and let σ be the substitution $[t := (t/t)/t; at_0 := ((t/t)/(t/t))/(t/t)]$. Then for all T, c in CAT_{AT}: $T \vdash_{\mathbf{L}} c$ iff $\sigma(T) \vdash_{\mathbf{L}} \sigma(c)$.

Proof: by induction on $d(T \vdash c)$, the degree of $T \vdash c$.

• $d(T \vdash c) = 0$. Then the categories T, c are members of the set $\text{AT} = \{t, at_0, at_1, \ldots, at_k\}$, while the categories $\sigma(T), \sigma(c)$ are members of the set $\text{AT}' = \{(t/t)/t, ((t/t)/(t/t))/(t/t), at_1, \ldots, at_k\}$, and the claim holds in view of the fact that both for $T, c \in \text{AT}$ and for $T, c \in \text{AT}'$ we have that $T \vdash_{\mathbf{L}} c$

entails that $T = c$. This is straightforward for $T, c \in \text{AT}$ (by at_i-count for $at_i \in \text{AT}$). For $T, c \in \text{AT}'$:

∘ If $T \vdash_{\mathbf{L}} at_j$ and $1 \leq j \leq k$, then for $c' \in \lfloor c_p \backslash ... \backslash c_1 \backslash at_j / c_{p+1} / .../ c_{p+q} \rfloor$: $T = T_1, \ldots, T_p, c', T_{p+q}, \ldots, T_{p+1}$ (and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$) by (53). The only member of AT′ in $\lfloor c_p \backslash ... \backslash c_1 \backslash at_j / c_{p+1} / .../ c_{p+q} \rfloor$ is at_j, and $at_j \in \lfloor at_j \rfloor$. Therefore, $p + q = 0$ and $T = at_j$.

∘ If $T \vdash_{\mathbf{L}} (t/t)/t$, then $T, t, t \vdash_{\mathbf{L}} t$ by (51), since $(t/t)/t \in \lfloor t/t/t \rfloor$. By (53), for $c' \in \lfloor c_p \backslash ... \backslash c_1 \backslash t / c_{p+1} / .../ c_{p+q} \rfloor$: $T, t, t = T_1, \ldots, T_p, c', T_{p+q}, \ldots, T_{p+1}$ and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$. For $c' \in \text{AT}'$, this entails (*i*) $c' = (t/t)/t$ and $c' \in \lfloor t/t/t \rfloor$; or (*ii*) $c' = ((t/t)/(t/t))/(t/t)$ and $c' \in \lfloor t/t/(t/t)/(t/t) \rfloor$. If (*ii*), then $T, t, t = ((t/t)/(t/t))/(t/t), S, t, t$ and $S, t, t = T''', T'', T'$, where $T''' \vdash_{\mathbf{L}} t/t$, $T'' \vdash_{\mathbf{L}} t/t$ and $T' \vdash_{\mathbf{L}} t$ – which is impossible by Corollary (1). So, assume (*i*). Then $T, t, t = (t/t)/t, S, t, t$ and $S, t, t = T'', T'$, where $T'' \vdash_{\mathbf{L}} t$ and $T' \vdash_{\mathbf{L}} t$ – which, by Corollary (3), entails that S is empty and, consequently, that $T = (t/t)/t$.

∘ If $T \vdash_{\mathbf{L}} ((t/t)/(t/t))/(t/t)$, then $T, t/t, t/t, t \vdash_{\mathbf{L}} t$ by (51), due to the fact that $c \in \lfloor t/t/(t/t)/(t/t) \rfloor$. By (53), for $c' \in \lfloor c_p \backslash ... \backslash c_1 \backslash t / c_{p+1} / .../ c_{p+q} \rfloor$: $T, t/t, t/t, t = T_1, \ldots, T_p, c', T_{p+q}, \ldots, T_{p+1}$ and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$, so that again (*i*) $c' = (t/t)/t$; or (*ii*) $c' = ((t/t)/(t/t))/(t/t)$. If (*i*), then $T, t/t, t/t, t = (t/t)/t, S, t/t, t/t, t$ and $S, t/t, t/t, t = T'', T'$, where $T'' \vdash_{\mathbf{L}} t$ and $T' \vdash_{\mathbf{L}} t$ – which is impossible by Corollary (2). So, assume (*ii*). Then $T, t/t, t/t, t = ((t/t)/(t/t))/(t/t), S, t/t, t/t, t$ and $S, t/t, t/t, t = T''', T'', T'$, where $T''' \vdash_{\mathbf{L}} t/t$, $T'' \vdash_{\mathbf{L}} t/t$ and $T' \vdash_{\mathbf{L}} t$ – which, by Corollary (4), entails that S is empty and, consequently, that $T = ((t/t)/(t/t))/(t/t)$.

• $d(T \vdash c) > 0$. If $c \in \text{CAT}_{\text{AT}}$ and $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash at / c_{p+1} / .../ c_{p+q} \rfloor$, then:
(A) $at \in \{at_1, \ldots, at_k\}$ and $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash at / \sigma(c_{p+1}) / .../ \sigma(c_{p+q}) \rfloor$;
(B) $at = t$ and $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/t/\sigma(c_{p+1}) / .../ \sigma(c_{p+q}) \rfloor$; or
(C) $at = at_0$ and $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/(t/t)/(t/t)/\sigma(c_{p+1}) / .../ \sigma(c_{p+q}) \rfloor$.
Since $p + q > 0$ or $p + q = 0$, six cases can be distinguished:

∘ $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash at_j / c_{p+1} / .../ c_{p+q} \rfloor$, $1 \leq j \leq k$, and $p + q > 0$:
$T \vdash_{\mathbf{L}} c$ iff$_1$ $c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1} \vdash_{\mathbf{L}} at_j$
iff$_2$ $\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}) \vdash_{\mathbf{L}} \sigma(at_j) =$
$\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}) \vdash_{\mathbf{L}} at_j =$

$\sigma(c_1), \ldots, \sigma(c_p), \sigma(T), \sigma(c_{p+q}) \ldots, \sigma(c_{p+1}) \vdash_{\mathbf{L}} at_j$
iff$_3$ $\sigma(T) \vdash_{\mathbf{L}} \sigma(c)$.

'iff$_1$' and 'iff$_3$' hold by (51) (since $c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash at_j / c_{p+1} / \ldots / c_{p+q} \rfloor$, while $\sigma(c) \in \lfloor \sigma(c_p) \backslash \ldots \backslash \sigma(c_1) \backslash at_j / \sigma(c_{p+1}) / \ldots / \sigma(c_{p+q}) \rfloor$ due to (A); and 'iff$_2$' holds by induction hypothesis ($d(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1} \vdash at_j) < d(T \vdash c)$, because $p + q > 0$).

∘ $c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash t / c_{p+1} / \ldots / c_{p+q} \rfloor$ and $p + q > 0$:
$T \vdash_{\mathbf{L}} c$ iff$_1$ $c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1} \vdash_{\mathbf{L}} t$
iff$_2$ $\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}) \vdash_{\mathbf{L}} \sigma(t) =$
$\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}) \vdash_{\mathbf{L}} (t/t)/t$
iff$_3$ $\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}), t, t \vdash_{\mathbf{L}} t =$
$\sigma(c_1), \ldots, \sigma(c_p), \sigma(T), \sigma(c_{p+q}) \ldots, \sigma(c_{p+1}), t, t \vdash_{\mathbf{L}} t$
iff$_4$ $\sigma(T) \vdash_{\mathbf{L}} \sigma(c)$.

'iff$_1$', iff$_3$' and iff$_4$' hold by (51) ($c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash t / c_{p+1} / \ldots / c_{p+q} \rfloor$, $(t/t)/t \in \lfloor t/t/t \rfloor$, and $\sigma(c) \in \lfloor \sigma(c_p) \backslash \ldots \backslash \sigma(c_1) \backslash t/t/t/\sigma(c_{p+1}) / \ldots / \sigma(c_{p+q}) \rfloor$ due to (B)); and 'iff$_2$' holds by induction hypothesis (since $p + q > 0$).

∘ $c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash at_0 / c_{p+1} / \ldots / c_{p+q} \rfloor$ and $p + q > 0$:
$T \vdash_{\mathbf{L}} c$ iff$_1$ $c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1} \vdash_{\mathbf{L}} at_0$
iff$_2$ $\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}) \vdash_{\mathbf{L}} \sigma(at_0) =$
$\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}) \vdash_{\mathbf{L}} ((t/t)/(t/t))/(t/t)$
iff$_3$ $\sigma(c_1, \ldots, c_p, T, c_{p+q} \ldots, c_{p+1}), t/t, t/t, t \vdash_{\mathbf{L}} t =$
$\sigma(c_1), \ldots, \sigma(c_p), \sigma(T), \sigma(c_{p+q}) \ldots, \sigma(c_{p+1}), t/t, t/t, t \vdash_{\mathbf{L}} t$
iff$_4$ $\sigma(T) \vdash_{\mathbf{L}} \sigma(c)$.

'iff$_1$', iff$_3$' and iff$_4$' hold by (51) (since $c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash at_0 / c_{p+1} / \ldots / c_{p+q} \rfloor$, $((t/t)/(t/t))/(t/t) \in \lfloor t/t/(t/t)/(t/t) \rfloor$ and – on account of (C) – $\sigma(c) \in \lfloor \sigma(c_p) \backslash \ldots \backslash \sigma(c_1) \backslash t/t/(t/t)/(t/t)/\sigma(c_{p+1}) / \ldots / \sigma(c_{p+q}) \rfloor$); and 'iff$_2$' holds by induction hypothesis (since $p + q > 0$).

∘ $c \in \lfloor at_j \rfloor$ and $1 \leq j \leq k$:
$T \vdash_{\mathbf{L}} at_j$ iff$_1$ for $c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash at_j / c_{p+1} / \ldots / c_{p+q} \rfloor$:
$T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$
and for all i, $1 \leq i \leq p + q$: $T_i \vdash_{\mathbf{L}} c_i$
iff$_2$ for $c \in \lfloor c_p \backslash \ldots \backslash c_1 \backslash at_j / c_{p+1} / \ldots / c_{p+q} \rfloor$:
$T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$
and for all i, $1 \leq i \leq p + q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$
iff$_3$ for $\sigma(c) \in \lfloor \sigma(c_p) \backslash \ldots \backslash \sigma(c_1) \backslash at_j / \sigma(c_{p+1}) / \ldots / \sigma(c_{p+q}) \rfloor$:
$\sigma(T) = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1})$

and for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$

iff$_4$ $\sigma(T) \vdash_{\mathbf{L}} at_j$.

Note that $at_j = \sigma(at_j)$, and that 'iff$_1$' holds by (53); 'iff$_2$' holds by induction hypothesis ($d(T \vdash_{\mathbf{L}} at_j) > 0$ entails that $p+q > 0$, hence $d(T_i \vdash_{\mathbf{L}} c_i) < d(T \vdash_{\mathbf{L}} at_j)$ for all i); 'iff$_3$' holds by (A); and 'iff$_4$' holds by (53).

$\circ$ $c \in \lfloor t \rfloor$:

$T \vdash_{\mathbf{L}} t$ iff$_1$ for $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash t / c_{p+1} / ... / c_{p+q} \rfloor$:

$T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$

and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$

iff$_2$ for $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash t / c_{p+1} / ... / c_{p+q} \rfloor$:

$T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$

and for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$

iff$_3$ for $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t / t / t / \sigma(c_{p+1}) / ... / \sigma(c_{p+q}) \rfloor$:

$\sigma(T) = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1})$

and for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$

iff$_4$ $\sigma(T), t, t \vdash_{\mathbf{L}} t$

iff$_5$ $\sigma(T) \vdash_{\mathbf{L}} (t/t)/t$.

Note that $(t/t)/t = \sigma(t)$, and that 'iff$_1$' holds by (53); 'iff$_2$' holds by induction hypothesis; 'iff$_3$' holds by (B); 'iff$_5$' holds by (51) (since $(t/t)/t \in \lfloor t/t/t \rfloor$); and the 'only if' part of 'iff$_4$' is an application of (53) (since $t \vdash_{\mathbf{L}} t$). As for the 'if' part of 'iff$_4$': if the final value of $\sigma(c)$ is t, then either

$$\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t / t / (t/t) / (t/t) / \sigma(c_{p+1}) / ... / \sigma(c_{p+q}) \rfloor \text{ or}$$
$$\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t / t / t / \sigma(c_{p+1}) / ... / \sigma(c_{p+q}) \rfloor.$$

Hence if $\sigma(T), t, t \vdash_{\mathbf{L}} t$, then, by (53), either

(*i*) for some $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t / t / (t/t) / (t/t) / \sigma(c_{p+1}) / ... / \sigma(c_{p+q}) \rfloor$:

- $\sigma(T), t, t = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1}), T''', T'', T'$,
- for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$, and
- $T''' \vdash_{\mathbf{L}} t/t$, $T'' \vdash_{\mathbf{L}} t/t$, and $T' \vdash_{\mathbf{L}} t$; or

(*ii*) for some $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t / t / t / \sigma(c_{p+1}) / ... / \sigma(c_{p+q}) \rfloor$:

- $\sigma(T), t, t = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1}), T'', T'$,
- for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$, and
- $T'' \vdash_{\mathbf{L}} t$ and $T' \vdash_{\mathbf{L}} t$.

However, (*i*) is impossible by Corollary (1), and Corollary (3) entails that (*ii*) is only possible if $T'' = T' = t$.

$\circ$ $c \in \lfloor at_0 \rfloor$:
$T \vdash_{\mathbf{L}} at_0$ iff_1 for $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash t / c_{p+1} / .../ c_{p+q} \rfloor$:
$T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$
and for all i, $1 \leq i \leq p+q$: $T_i \vdash_{\mathbf{L}} c_i$
iff_2 for $c \in \lfloor c_p \backslash ... \backslash c_1 \backslash at_0 / c_{p+1} / .../ c_{p+q} \rfloor$:
$T = T_1, \ldots, T_p, c, T_{p+q}, \ldots, T_{p+1}$
and for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$
iff_3 for $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/(t/t)/(t/t)/\sigma(c_{p+1})/.../\sigma(c_{p+q}) \rfloor$:
$\sigma(T) = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1})$
and for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$
iff_4 $\sigma(T), t/t, t/t, t \vdash_{\mathbf{L}} t$
iff_5 $\sigma(T) \vdash_{\mathbf{L}} ((t/t)/(t/t))/(t/t)$.

Note that $((t/t)/(t/t))/(t/t) = \sigma(at_0)$, and that '$\text{iff}_1$' holds by (53); '$\text{iff}_2$' holds by induction hypothesis; 'iff_3' holds by (C); 'iff_5' holds by (51) (since $((t/t)/(t/t))/(t/t) \in \lfloor t/t/(t/t)/(t/t) \rfloor$); and the 'only if' part of 'iff_4' is an application of (53) (since $t/t \vdash_{\mathbf{L}} t/t$ and $t \vdash_{\mathbf{L}} t$). As for the 'if' part of 'iff_4': again, if the final value of $\sigma(c)$ is t, then either

$$\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/(t/t)/(t/t)/\sigma(c_{p+1})/.../\sigma(c_{p+q}) \rfloor \text{ or}$$
$$\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/t/\sigma(c_{p+1})/.../\sigma(c_{p+q}) \rfloor.$$

Hence if $\sigma(T), t/t, t/t, t \vdash_{\mathbf{L}} t$, then, by (53), either
(*i*) for some $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/t/\sigma(c_{p+1})/.../\sigma(c_{p+q}) \rfloor$:
– $\sigma(T), t/t, t/t, t = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1}), T'', T'$,
– for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$, and
– $T'' \vdash_{\mathbf{L}} t$ and $T' \vdash_{\mathbf{L}} t$; or
(*ii*) for some $\sigma(c) \in \lfloor \sigma(c_p) \backslash ... \backslash \sigma(c_1) \backslash t/t/(t/t)/(t/t)/\sigma(c_{p+1})/.../\sigma(c_{p+q}) \rfloor$:
– $\sigma(T), t/t, t/t, t = \sigma(T_1), \ldots, \sigma(T_p), \sigma(c), \sigma(T_{p+q}), \ldots, \sigma(T_{p+1}), T''', T'', T'$,
– for all i, $1 \leq i \leq p+q$: $\sigma(T_i) \vdash_{\mathbf{L}} \sigma(c_i)$, and
– $T''' \vdash_{\mathbf{L}} t/t$, $T'' \vdash_{\mathbf{L}} t/t$, and $T' \vdash_{\mathbf{L}} t$.

This time, (*i*) is impossible by Corollary (2), and Corollary (4) entails that (*ii*) is only possible if $T''' = T'' = t/t$ and $T' = t$. □

Finally, the following claim shows that one can generalize the substitution used in Claim 1 to encode any finite number of atomic categories. Let, for $c \in \text{CAT}$ and $n \in \mathbb{N}$:

$$\beta(c) = ((c/c)/(c/c))/(c/c);$$
$$\alpha(c) = (c/c)/c;$$

$\alpha^0(c) = c$; and
$\alpha^{n+1}(c) = \alpha^n(\alpha(c))$.

Claim 2: Let $A = \langle t, at_1, \ldots, at_m \rangle$ be a sequence of distinct atomic categories such that $m \geq 1$, and let σ_A be the substitution

$$[t := \alpha^m(t); at_1 := \beta(\alpha^{m-1}(t)); \ldots; at_m := \beta(\alpha^{m-m}(t))].$$

Then for all T, c in $\text{CAT}_{\{t, at_1, \ldots, at_m\}}$: $T \vdash_{\mathbf{L}} c$ iff $\sigma_A(T) \vdash_{\mathbf{L}} \sigma_A(C)$.

Proof: by induction on m.

• $m = 1$. Then Claim 2 comes down to Claim 1 (with at_0 and k instantiated as at_1 and 0, respectively).

• $m > 1$. Observe (i) that $\sigma_A(c) = \sigma'_A(\sigma''_A(c))$ for the substitutions $\sigma'_A = [t := \alpha^{m-1}(t); at_1 := \beta(\alpha^{(m-1)-1}(t)); \ldots; at_{m-1} := \beta(\alpha^{(m-1)-(m-1)}(t))]$ and $\sigma''_A = [t := \alpha(t); at_m := \beta(t)]$; and (ii) that $\sigma'_A(c) = \sigma_{A'}(c)$ for the sequence $A' = \langle t, at_1, \ldots, at_{m-1} \rangle$. Consequently, we have the following equivalences:
$\sigma_A(T) \vdash_{\mathbf{L}} \sigma_A(c)$ iff$_1$ $\sigma'_A(\sigma''_A(T)) \vdash_{\mathbf{L}} \sigma'_A(\sigma''_A(c))$
iff$_2$ $\sigma''_A(T) \vdash_{\mathbf{L}} \sigma''_A(c)$
iff$_3$ $T \vdash_{\mathbf{L}} c$.
'iff$_1$' holds by observation (i); 'iff$_2$' holds by induction hypothesis and observation (ii) (note that $m - 1 < m$, and that $\sigma''_A(c) \in \text{CAT}_{\{t, at_1, \ldots, at_{m-1}\}}$ if $c \in \text{CAT}_{\{t, at_1, \ldots, at_m\}}$); and 'iff$_3$' is another application of Claim 1 (with at_0 and k instantiated as at_m and $m - 1$, respectively). □

References

Abusch, D. (1993). 'The Scope of Indefinites'. In J. Groenendijk (ed.) (1993), *Plurals and Anaphora*. ESPRIT Basic Research Project 6852, Dynamic Interpretation of Natural Language, DYANA-2 Deliverable R2.2A, Part I.

Adj (J.A. Goguen, J.W. Thatcher, E.G. Wagner, J.B. Wright) (1977). 'Initial Algebra Semantics and Continuous Algebras'. *Journal of the Association for Computing Machinery* **24**, 68–95.

Ajdukiewicz, K. (1935). 'Die syntaktische Konnexität'. *Studia Philosophica* **1**, 1–27.

Akmajian, A. (1977). 'The Complement Structure of Perception Verbs in an Autonomous Syntax Framework'. In Culicover, Wasow and Akmajian (eds.) (1977).

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.

Auster, P. (1988). *The New York Trilogy. City of Glass / Ghosts / The Locked Room*. Faber and Faber, London and Boston.

Bach, E. (1980). 'Tenses and Aspects as Functions on Verb-Phrases'. In C. Rohrer (ed.) (1980), *Time, Tense and Quantifiers*. Max Niemeyer, Tübingen.

Barendregt, H. (1981). *The Lambda Calculus. Its Syntax and Semantics*. North Holland, Amsterdam.

Bar-Hillel, Y. (1953). 'A Quasi-Arithmetical Notation for Syntactic Description.' *Language* **29**, 47–58. Reprinted in Bar-Hillel (ed.) (1964).

Bar-Hillel, Y. (ed.) (1964). *Language and Information*. Addison-Wesley, Reading (Massachusetts).

Bar-Hillel, Y., C. Gaifman, and E. Shamir (1960). 'On Categorial and Phrase Structure Grammars'. *The Bulletin of the Research Council of Israel* **9**, 1–16. Reprinted in Bar-Hillel (ed.) (1964).

Barry, G., and G. Morrill (eds.) (1990). *Studies in Categorial Grammar. Edinburgh Working Papers in Cognitive Science 5*. Centre for Cognitive Science, Edinburgh.

Barry, G., and M. Pickering (1990). 'Dependency and Constituency in Categorial Grammar'. In Barry and Morrill (eds.) (1990). Also in A. Lecomte (ed.)

(1992), *Word Order in Categorial Grammar / L'Ordre des Mots dans les Grammaires Catégorielles*. Editions Adosa, Clermont-Ferrand.

Barwise, J. (1981). 'Scenes and Other Situations'. *Journal of Philosophy* **78**, 369–397.

Barwise, J., and J. Perry (1983). *Situations and Attitudes*. MIT Press, Cambridge (Massachusetts).

Benthem, J. van (1980). 'Universal Algebra and Model Theory. Two Excursions on the Border.' Report ZW–7908. Department of Mathematics, Groningen University.

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

Benthem, J. van (1988). 'The Lambek calculus'. In Oehrle, Bach and Wheeler (eds.) (1988).

Benthem, J. van (1989). 'Polyadic Quantifiers'. *Linguistics and Philosophy* **12**, 437–464.

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

Bouma, G. (1989). 'Structural Completeness'. Manuscript Research Institute for Knowledge Systems, Maastricht.

Buszkowski, W. (1986). 'Completeness Results for Lambek Syntactic Calculus'. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **32**, 13–28.

Buszkowski, W., W. Marciszewski and J. van Benthem (eds.) (1988). *Categorial Grammar*. John Benjamins, Amsterdam.

Chomsky, N. (1981). *Lectures on Government and Binding*. Foris, Dordrecht.

Church, A. (1932/3). 'A Set of Postulates for the Foundation of Logic'. *Annals of Mathematics* **33**, 346–366 and **34**, 839–864.

Cooper, R. (1975). *Montague's Semantic Theory and Transformational Syntax*. Dissertation University of Massachusetts at Amherst.

Cooper, R. (1979). 'Variable Binding and Relative Clauses'. In F. Guenthner and S. Schmidt (eds.) (1979), *Formal Semantics and Pragmatics for Natural Languages*. Reidel, Dordrecht.

Cooper, R. (1983). *Quantification and Syntactic Theory*. Reidel, Dordrecht.

Cooper, R., and T. Parsons (1976). 'Montague Grammar, Generative Semantics, and Interpretive Semantics'. In Partee (ed.) (1976).

Culicover, P., T. Wasow and A. Akmajian (eds.) (1977). *Formal Syntax*. Academic Press, New York.

Davidson, D., and G. Harman (eds.) (1972). *Semantics of Natural Language*. Reidel, Dordrecht.

Dekker, P. (1990). 'The Scope of Negation in Discourse. Towards a Flexible Dynamic Montague Grammar'. ITLI Prepublication Series LP 90-09, Institute for Language, Logic and Information, University of Amsterdam. Also in Groenendijk, Stokhof and Beaver (eds.) (1991).

Dekker, P. (1993). *Transsentential Meditations. Ups and Downs in Dynamic Semantics*. Dissertation University of Amsterdam.

Diels, H., and W. Kranz (eds.) (1952). *Die Fragmente der Vorsokratiker*. Berlin.

Does, J. van der (1990). 'A Generalized Quantifier Logic for Naked Infinitives'. *Linguistics and Philosophy* **14**, 241–294.

Does, J. van der (1992). *Applied Quantifer Logics. Collectives / Naked Infinitives*. Dissertation University of Amsterdam.

Dosen, K. (1985). 'A Completeness Theorem for the Lambek Calculus of Syntactic Categories'. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **31**, 235–241.

Dowty, D. (1985). 'On Recent Analyses of the Semantics of Control'. *Linguistics and Philosophy* **8**, 291–331.

Dowty, D. (1988). 'Type Raising, Functional Composition, and Non-Constituent Conjunction'. In Oehrle, Bach and Wheeler (eds.) (1988).

Dowty, D., R. Wall and S. Peters (1981). *Introduction to Montague Semantics*. Reidel, Dordrecht.

Dummett, M. (1976). *Frege. Philosophy of Language*. Duckworth, London.

Eijck, J. van (1991). 'Quantification'. In D. Wunderlich and A. von Stechow (eds.) (1991), *Semantics. An International Handbook of Contemporary Research*. De Gruyter, Berlin and New York.

Emms, M. (1990). 'Polymorphic Quantifiers'. In Barry and Morrill (eds.) (1990), and (abridged) in Stokhof and Torenvliet (eds.) (1990).

Emms, M. (1992). *Logical Ambiguity*. Dissertation University of Edinburgh.

Evans, G. (1980). 'Pronouns'. *Linguistic Inquiry* **11**, 337–362.

Faverey, H. (1977). *Chrysanten, roeiers*. De Bezige Bij, Amsterdam.

Faverey, H. (1984). *Chrysanthemums, rowers*. Translated by J.M. Coetzee. In J.S. Holmes and W.J. Smith (eds.), *Dutch Interior. Postwar Poetry of the Netherlands and Flanders*. Columbia University Press, New York.

Fodor, J., and I. Sag (1982). 'Referential and Quantificational Indefinites'. *Linguistics and Philosophy* **5**, 355–398.

Frege, G. (1884). *Die Grundlagen der arithmetik. Eine logisch-mathematische Untersuchung über den Begriff der Zahl.* Köbner, Breslau. Reprint (1961) by Georg Olms, Hildesheim.

Gallin, D. (1975). *Intensional and Higher-Order Modal Logic.* North Holland, Amsterdam.

Gamut, L. (1991). *Logic, Language and Meaning. Volume I: Introduction to Logic. Volume II: Intensional Logic and Logical Grammar.* University of Chicago Press, Chicago and London.

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.* Basil Blackwell, Oxford.

Geach, P. (1962). *Reference and Generality. An Examination of Some Medieval and Modern Theories.* Cornell University Press, Ithaca and London.

Gentzen, G. (1934). 'Untersuchungen über das logische Schließen'. *Mathematische Zeitschrift* **39**, 176–210 and 405–431.

Girard, J.-Y. (1987). 'Linear Logic'. *Theoretical Computer Science* **50**, 1–102.

Groenendijk, J., T. Janssen and M. Stokhof (eds.) (1981). *Formal Methods in the Study of Language.* Mathematical Centre, Amsterdam.

Groenendijk, J., T. Janssen, and M. Stokhof (eds.) (1984). *Truth, Interpretation and Information. Selected Papers from the Third Amsterdam Colloquium.* Foris, Dordrecht.

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 (1989). 'Type-shifting Rules and the Semantics of Interrogatives'. In G. Chierchia, B. Partee and R. Turner (eds.) (1989), *Properties, Types and Meaning. Volume II: Semantic Issues.* Kluwer, Dordrecht.

Groenendijk, J., and M. Stokhof (1990). 'Dynamic Montague Grammar'. In L. Kalmán and L. Pólos (eds.) (1990), *Papers from the Second Symposium on Logic and Language.* Akadémiai Kiadó, Budapest. Also in Groenendijk, Stokhof, and Beaver (eds.) (1991), 1–36.

Groenendijk, J., and M. Stokhof (1991). 'Dynamic Predicate Logic'. *Linguistics and Philosophy* **14**, 39–100.

Groenendijk, J., M. Stokhof, and D. Beaver (eds.) (1991). *Quantification and Anaphora I.* ESPRIT Basic Research Project 3175, Dynamic Interpretation of Natural Language, DYANA Deliverable R2.2A.

Halvorsen, P.-K., and W. Ladusaw (1979). 'Montague's "Universal Grammar": an Introduction for the Linguist'. *Linguistics and Philosophy* **3**, 185–223.

Heijenoort, J. van (ed.) (1967). *From Frege to Gödel: A Source Book in Mathematical Logic 1879–1931*. Harvard University Press, Cambridge (Massachusetts).

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*. Dissertation University of Massachusetts at Amherst.

Hendriks, H. (1988). 'Type Change in Semantics: the Scope of Quantification and Coordination'. In Klein and Van Benthem (eds.) (1988).

Henkin, L. (1950). 'Completeness in the Theory of Types'. *Journal of Symbolic Logic* **15**, 81–91.

Henkin, L. (1963). 'A Theory of Propositional Types'. *Fundamenta Mathematicae* **52**, 323–344.

Hepple, M. (1990). *The Grammar and Processing of Order and Dependency: a Categorial Approach*. Dissertation University of Edinburgh.

Hintikka, J. (1974). 'Quantifiers vs. Quantification Theory'. *Linguistic Inquiry* **5**, 153–177.

Jacobs, R., and P. Rosenbaum (eds.) (1971). *Readings in English Transformational Grammar*. Blaisdell, Waltham (Massachusetts).

Janssen, T. (1984). 'Individual Concepts are Useful'. In Landman and Veltman (eds.) (1984).

Janssen, T. (1986). *Foundations and Applications of Montague Grammar. Part 1: Philosophy, Framework, Computer Science. Part 2: Applications to Natural Language*. CWI Tracts 19 and 28, Amsterdam.

Kamp, H. (1981). 'A Theory of Truth and Semantic Representation'. In Groenendijk, Janssen and Stokhof (eds.) (1981). Reprinted in Groenendijk, Janssen and Stokhof (eds.) (1984).

Karttunen, L. (1968a). *What Do Referential Indices Refer to?*. RAND Corporation Publication P-3854. Santa Monica (California).

Karttunen, L. (1968b). *What makes Definite Noun Phrases Definite?*. RAND Corporation Publication P-3871. Santa Monica (California).

Karttunen, L. (1969). 'Pronouns and Variables'. In R. Binnick *et al.* (eds.) (1969), *Papers from the Fifth Regional Meeting of the Chicago Linguistic Society'*. Department of Linguistics, University of Chicago.

Keenan, E., and L. Faltz (1985). *Boolean Semantics for Natural Language*. Reidel, Dordrecht.

Keenan, E. (1987). 'Unreducible n-ary Quantifiers in Natural Language'. In

Gärdenfors, P. (ed.) (1987), *Generalized Quantifiers. Linguistic and Logical Approaches*. Reidel, Dordrecht.

Keenan, E. (1992). 'Beyond the Frege Boundary'. *Linguistic and Philosophy* **15**, 199–221.

Klein, E. and J. van Benthem (eds.) (1988). *Categories, Polymorphism and Unification*, Centre for Cognitive Science, University of Edinburgh, and Institute for Language, Logic and Information, University of Amsterdam.

Klein, E., and I. Sag (1985). 'Type-Driven Translation'. *Linguistics and Philosophy* **8**, 163–202.

König, E. (1989). 'Parsing as Natural Deduction'. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*. Vancouver.

Lakoff, G. (1971). 'Generative Semantics'. In Jacobs and Rosenbaum (eds.) (1971).

Lakoff, G. (1972). 'Linguistics and Natural Logic'. In Davidson and Harman (eds.) (1972).

Lambek, J. (1958). 'The Mathematics of Sentence Structure'. *American Mathematical Monthly* **65**, 154–169. Reprinted in Buszkowski, Marciszewski and Van Benthem (eds.) (1988).

Lambek, J. (1990). 'Logic without structural rules. (Another Look at Cut Elimination)'. Manuscript McGill University, Montreal.

Landman, F., and I. Moerdijk (1983). 'Compositionality and the Analysis of Anaphora'. *Linguistics and Philosophy* **6**, 89-114.

Landman, F., and F. Veltman (eds.) (1984). *Varieties of Formal Semantics. Proceedings of the Fourth Amsterdam Colloquium*. Foris, Dordrecht.

Leslie, N. (1990). 'Contrasting Styles of Categorial Derivations'. In Barry and Morrill (eds.) (1990).

Lewis, D. (1974). 'Tensions'. In M. Munitz and P. Unger (eds.) (1974), *Semantics and Philosophy*. New York University Press, New York.

May, R. (1985). *Logical Form: Its Structure and Derivation*. MIT Press, Cambridge (Massachusetts).

Montague, R. (1970a). 'English as a Formal Language'. In B. Visentini *et al.* (eds.) (1970), *Linguaggi nella Società e nella Tecnica*. Edizioni di Communità, Milan. Page references concern the reprint as Chapter 6 of Thomason (ed.) (1974).

Montague, R. (1970b). 'Universal Grammar'. *Theoria* **36**, 373–398. Page references concern the reprint as Chapter 7 of Thomason (ed.) (1974).

Montague, R. (1973). 'The Proper Treatment of Quantification in Ordinary English'. In J. Hintikka, J. Moravcsik and P. Suppes (eds.) (1973), *Approaches to*

Natural Language. Proceedings of the 1970 Stanford Workshop on Grammar and Semantics. Reidel, Dordrecht. Page references concern the reprint as Chapter 8 of Thomason (ed.) (1974).

Montague, R. (1974). See Thomason (ed.) (1974).

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

Moortgat, M. (1990). 'Unambiguous Proof Representations for the Lambek Calculus'. In Stokhof and Torenvliet (eds.) (1990).

Moortgat, M. (1993). 'Generalized Quantifiation and Discontinuous Type Constructors'. In W. Sijtsma and A. van Horck (eds.) (1993), *Proceedings of the Tilburg Symposium on Discontinuous Dependencies*. De Gruyter, Berlin.

Moortgat, M., and G. Morrill (1992). 'Heads and Phrases: Type Calculus for Dependency and CoNstituency'. To appear in *Journal of Language, Logic and Information*.

Morrill, G. (1990a). 'Intensionality and Boundedness.' *Linguistics and Philosophy* **13**, 699–726.

Morrill, G. (1990b). 'Grammar and Logical Types.' In Barry and Morrill (eds.) (1990). Also in Stokhof and Torenvliet (eds.) (1990).

Morrill, G. (1992). *Type-Logical Grammar*. Research Report, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, and Onderzoeksinstituut voor Taal en Spraak, Universiteit Utrecht. Monograph to appear in *Studies in Language, Logic and Information*. Kluwer, Dordrecht.

Morrill, G. (1993). 'Discontinuity and Pied-Piping in Categorial Grammar'. In M. Moortgat (ed.) (1993), *Polymorphic Treatments*. ESPRIT Basic Research Project 6852, Dynamic Interpretation of Natural Language, DYANA-2 Deliverable R1.3A.

Muskens, R. (1989). *Meaning and Partiality*. Dissertation University of Amsterdam.

Oehrle, R., E. Bach and D. Wheeler (eds.) (1988). *Categorial Grammars and Natural Language Structures*. Reidel, Dordrecht.

Orey, S. (1959). 'Model Theory for the Higher Order Predicate Calculus', *Transactions of the American Mathematical Society* **92**, 72–84.

Partee, B. (1970). 'Opacity, Coreference, and Pronouns'. *Synthese* **21**, 359–385. Also in Davidson and Harman (eds.) (1972).

Partee, B. (ed.) (1976). *Montague Grammar*. Academic Press, New York.

Partee, B. (1984). 'Compositionality'. In Landman and Veltman (eds.) (1984).

Partee, B. (1986). 'Noun Phrase Interpretation and Type-shifting Principles'. In J.

Groenendijk, D. de Jongh and M. Stokhof (eds.) (1986), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*. Foris, Dordrecht.

Partee, B., and M. Rooth (1983). 'Generalized Conjunction and Type Ambiguity'. In R. Bäuerle, C. Schwarze and A. von Stechow (eds.) (1983), *Meaning, Use and Interpretation of Language*. De Gruyter, Berlin.

Pentus, M., (1992). 'Lambek Grammars are Context-Free'. Manuscript Department of Mathematical Logic, Moscow University.

Pollard, C., and I. Sag (1987). *Information-Based Syntax and Semantics. Volume 1. Fundamentals.* CSLI Lecture Notes, Stanford.

Ponse, A. (1988). 'Encoding Types in the Lambek Calculus'. In Klein and Van Benthem (eds.) (1988).

Prüst, H. (1992). *On Discourse Structuring, VP Anaphora and Gapping.* Dissertation University of Amsterdam.

Reinhart, T. (1976). *The Syntactic Domain of Anaphora.* Dissertation Massachusetts Institute of Technology, Cambridge (Massachusetts).

Reinhart, T. (1983). 'Coreference and Bound Anaphora: a Restatement of the Anaphora Questions'. *Linguistics and Philosophy* **6**, 47-88.

Rodman, R. (1976). 'Scope Phenomena, "Movement Transformations", and Relative Clauses'. In Partee (ed.) (1976).

Roorda, D (1991). *Resource Logics: Proof-theoretical Investigations.* Dissertation University of Amsterdam.

Rooth, M. (1985). *Association with Focus.* Dissertation University of Massachusetts at Amherst.

Rooth, M., and B. Partee (1982). 'Conjunction, Type Ambiguity, and Wide Sope "or" '. In D. Flickinger, M. Macken and N. Wiegand (eds.) (1982), *Proceedings of the 1982 West Coast Conference on Formal Linguistics.* Stanford Linguistics Department, Stanford.

Ross, J. (1967). *Constraints on Variables in Syntax.* Dissertation Massachusetts Institute of Technology, Cambridge (Massachusetts).

Sag, I. (1977). *Deletion and Logical Form.* Dissertation University of Pennsylvania.

Scha, R. (1981). 'Distributive, Collective and Cumulative Quantification'. In Groenendijk, Janssen and Stokhof (eds.) (1981). Reprinted in Groenendijk, Janssen and Stokhof (eds.) (1984).

Schönfinkel, M. (1924). 'Über die Bausteine der mathematischen Logik.' *Mathematische Annalen* **92**, 305–316.

Seuren, P. (1985). *Discourse Semantics.* Blackwell, Oxford.

Stechow, A. von (1974). 'ϵ–λ kontextfreie Sprachen: Ein Beitrag zu einer natürlichen formalen Semantik'. *Linguistische Berichte* **34**, 1–34.

Stokhof, M., and L. Torenvliet (eds.) (1990). *Proceedings of the Seventh Amsterdam Colloquium*. Institute for Language, Logic and Information, University of Amsterdam.

Thomason, R. (ed.) (1974). *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven.

Twain, M. (1992). *Tom Sawyer and Huckleberry Finn*. Wordsworth Editions, Ware (Hertfordshire).

Williams, E. (1977). 'Discourse and Logical Form'. *Linguistic Inquiry* **8**, 101–139.

Williams, E. (1983). 'Against Small Clauses'. *Linguistic Inquiry* **14**, 287–308.

Zielonka, W. (1981). 'Axiomatizability of Ajdukiewicz-Lambek Calculus by means of Cancellation Schemes'. *Zeitschrift fur mathemätische Logik und Grundlagen der Mathematik* **27**, 215–224.

Samenvatting

In formele-grammaticamodellen gaat men er gewoonlijk van uit dat uitdrukkingen tot syntactische *categorieën* behoren, en dat betekenissen van uitdrukkingen semantische *typen* bevolken. Dit proefschrift gaat over de gevolgen van deze dubbele veronderstelling.

Een eerste samenkomst van syntactische categorieën en semantische typen kan worden aangetroffen in de zeer invloedrijke logisch-linguïstische artikelen van Richard Montague, waar wiskundig geïnspireerde typentheorieën de semantiek leveren voor categoriaal-syntactisch gedefinieerde fragmenten van het Engels. Elke uitdrukking in deze fragmenten behoort tot een syntactische categorie, en heeft een betekenis van een semantisch type dat geheel bepaald wordt door de categorie waartoe de uitdrukking behoort. De betekenissen van de uitdrukkingen van een bepaalde categorie zijn derhalve steevast van het unieke bij die categorie behorende type: de toekenning van typen aan categorieën is met andere woorden een *functie*. Als gevolg hiervan is men genoodzaakt te generaliseren naar 'het ergste geval', want *alle* uitdrukkingen van een syntactische categorie dienen op uniforme wijze te worden voorzien van een betekenis van het meest ingewikkelde type dat voor *enige* uitdrukking van de categorie benodigd is.

Dit aspect van 'orthodoxe' Montague-grammatica is door onder anderen Barbara Partee en Mats Rooth bekritiseerd. Zij betogen dat de beschikbaarheid van bereiksambiguïteiten aantoont dat generaliseren naar het ergste geval niet altijd het juiste resultaat geeft. Bereiksambiguïteiten zijn dubbelzinnigheden die niet berusten op een verschil in lexicale betekenis of syntactische structuur, maar kunnen worden uitgelegd in termen van het onderlinge bereik van semantische operatoren, zoals bijvoorbeeld de operatoren die de betekenissen vormen van de uitdrukkingen *een handelsreiziger* en *elk gehucht* in de zin *Een handelsreiziger bezocht elk gehucht*, die immers kan worden uitgelegd als: 'er is een handelsreiziger die elk gehucht bezocht', maar ook als: 'elk gehucht werd door een handelsreiziger bezocht'.

Montague verantwoordde dergelijke dubbelzinnigheden in overeenstemming met het principe van compositionaliteit van betekenis, dat stelt dat de betekenis van een uitdrukking geheel bepaald wordt door de betekenissen van de samenstellende delen en van de manier waarop deze delen syntac-

tisch zijn gecombineerd. Wanneer een ambiguïteit niet gepaard gaat met een verschil in woordbetekenis of syntactische structuur zijn er blijkbaar verschillende constructiewijzen van één en dezelfde structuur in het geding. Voor de analyse van bereiksambiguïteiten (en de beregeling van anaforische pronomina) introduceerde Montague daarom zogenaamde syntactische variabelen en kwantificatieregels. Het feit echter dat deze hulpmiddelen nogal gekunstelde syntactische constructies opleveren heeft binnen het kader van de modeltheoretische semantiek de aanzet gegeven tot alternatieve benaderingen. Zo stelde Robin Cooper een opslagmechanisme voor kwantoren en pronomina voor dat weliswaar de minder prettige syntactische aspecten van kwantificatieregels omzeilt, maar anderzijds verstrikt raakt in semantische complicaties.

In Hoofdstuk 1 van dit proefschrift worden beide benaderingen besproken. Ook wordt onderzocht in hoeverre het mogelijk is om bereiksambiguïteiten in flexibele categoriale grammatica's te verantwoorden. Het blijkt dat – anders dan wel gesuggereerd is – bestaande stelsels op dit punt niet voldoen. In plaats daarvan wordt flexibele interpretatie voorgesteld als een alternatief dat de tegenintuïtieve syntactische en semantische aspecten van kwantificatieregels en opslag vermijdt en zo bijdraagt tot een betere taakverdeling tussen de syntactische en semantische component.

Flexibele interpretatie houdt in dat aan uitdrukkingen van de taal verzamelingen van op systematische wijze aan elkaar gerelateerde betekenissen van verschillende typen worden toegekend. In Hoofdstuk 1 wordt betoogd dat dit opgeven van de strategie van het generaliseren naar het ergste geval een goede zaak is – niet omdat het ergste geval niet altijd van toepassing is, maar omdat het gewoonweg niet bestaat. Het kan altijd erger.

Het flexibele alternatief behelst een omgekeerde strategie, die op lexicaal niveau naar 'het beste geval' generaliseert, want elke lexicale uitdrukking krijgt een betekenis van het minimale type dat voor die uitdrukking beschikbaar is. Gegeneraliseerde syntactisch/semantische regels zorgen voor de samenvoeging van alle betekenissen die 'in elkaar passen' en type-veranderende regels vormen afgeleide betekenissen uit lexicale en samengestelde basisbetekenissen, waarbij het recursieve karakter van deze regels het empirische gegeven weerspiegelt dat er geen ergste geval is. Met deze semantische flexibiliteit kan men bereiksambiguïteiten van kwantificerende en coördinerende uidrukkingen binnen de semantiek verantwoorden zonder ze ook als syntactische ambiguïteiten op te hoeven vatten. Bovendien blijken de empirische voorspellingen van deze 'flexibele Montague-grammatica' beter dan die van het standaardmodel.

In Hoofdstuk 2 van dit proefschrift wordt nagegaan of flexibele interpretatie een liberalisering van het principe van compositionaliteit van betekenis noodzakelijk maakt. Het uitgangspunt is Theo Janssens algebraïsche formalisering van dit principe. Een gedetailleerde beschouwing van de gevolgen van het compositionaliteitsprincipe voor de algemene organisatie van grammaticafragmenten leidt tot een formulering van het principle die enigszins afwijkt van die van Janssen. Betoogd wordt dat diens (equivalentie-)relatie van *similariteit* tussen veelsoortige algebra's vervangen dient te worden door een zwakkere (transitieve en reflexieve) relatie van *interpreteerbaarheid*. Deze is beter te motiveren en toe te passen en vermijdt lastige technische complicaties in de zogenaamde theorie van vertaling. In het licht van deze herziene formalisering blijkt dat ook in aanwezigheid van flexibele type-veranderende interpretatieregels het compositionaliteitsprincipe onverkort kan worden gehandhaafd in zijn meest 'intuïtieve' interpretatie, mits deze regels worden opgevat als unaire, syntactisch triviale operaties in de grammaticale algebra.

Hoofdstuk 3 van dit proefschrift gaat over de semantiek van de syntactische calculus van Joachim Lambek, een toonaangevend stelsel van flexibele categoriale grammatica dat in Hoofdstuk 1 ook al aan de orde kwam. De Lambek-calculus herleidt grammaticale ontleding tot logische bewijsvoering door een algemene notie van afleidbaarheid te definiëren waarmee kan worden vastgesteld of een reeks categorieën tot een bepaalde categorie behoort. Lambek formuleerde tevens een algoritme voor *Snede*-eliminatie, waarmee hij de beslisbaarheid van de calculus bewees: de vraag of een reeks categorieën tot een bepaalde categorie behoort, kan altijd in een eindig aantal stappen worden beantwoord.

Functie/argumentstructuur staat centraal in het categoriale grammaticamodel; het is dan ook een formalisme dat plaats kan bieden aan zowel syntactische categorieën als semantische typen. Johan van Benthem definieerde een typenlogische semantiek voor de Lambek-calculus waarin de interpretaties van een grammaticaal correcte uitdrukking direct bepaald worden door de afleidingen die de syntactische calculus voor deze uitdrukking levert. Hoofdstuk 3 presenteert en motiveert een licht gewijzigde formulering van deze semantiek, waarmee op eenvoudige wijze wordt bewezen da toepassing van Lambeks *Snede*-eliminatie-algoritme altijd resulteert in een afleiding die semantisch equivalent is met de oorspronkelijke afleiding. Dientengevolge is de Lambek-calculus ook semantisch beslisbaar: de interpretaties van een uitdrukking kunnen in een eindig aantal stappen worden gespecificeerd. De calculus kan bovendien verder genormaliseerd worden tot

een calculus die het probleem oplost dat verschillende *Snede*-vrije afleidingen van een uitdrukking dezelfde semantische interpretatie kunnen hebben: in de genormaliseerde calculus correspondeert elke interpretatie met precies één afleiding. Tenslotte kan met behulp van deze calculus de equivalentie van enkele andere normalisatiestrategieën – de bewijsnetten van Dirk Roorda en een compilatie-normalisatie van Michael Moortgat – worden aangetoond en een reductiestelling betreffende basiscategorieën worden bewezen.

Previous titles in the ILLC Dissertation Series

Paul Dekker
Transsentential Meditations. Ups and downs in dynamic semantics
ILLC Dissertation Series 1993–1

Harry Buhrman
Resource Bounded Reductions
ILLC Dissertation Series 1993–2

Rineke Verbrugge
Efficient Metamathematics
ILLC Dissertation Series 1993–3

Maarten de Rijke
Extending Modal Logic
ILLC Dissertation Series 1993–4