Institute for Language, Logic and Information

THE SCOPE OF NEGATION IN DISCOURSE

TOWARDS A FLEXIBLE DYNAMIC MONTAGUE GRAMMAR

Paul Dekker

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language LP-90-09

University of Amsterdam

The ITLI Prepublication Series

1986
86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas — A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem — Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens — A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh — Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem — Logical Syntax

1987
87-01 Jeroen Groenendijk, Martin Stokhof — Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch — Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer — Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem — Polyadic quantifiers
87-05 Víctor Sánchez Valencia — Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen — Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem — Categorial Grammar and Type Theory
87-08 Renate Bartsch — The Construction of Properties under Perspectives
87-09 Herman Hendriks — Type Change in Semantics: The Scope of Quantification and Coordination

1988
Logic, Semantics and Philosophy of Language:
LP-88-01 Michiel van Lambalgen — Algorithmic Information Theory
LP-88-02 Yde Venema — Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 — Year Report 1987
LP-88-04 Reinhard Muskens — Going partial in Montague Grammar
LP-88-05 Johan van Benthem — Logical Constants across Varying Types
LP-88-06 Johan van Benthem — Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch — Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof — Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen — A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe — A Blissymbolics Translation Program

Mathematical Logic and Foundations:
ML-88-01 Jaap van Oosten — Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen — The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman — Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra — On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:
CT-88-01 Ming Li, Paul M.B.Vitanyi — Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid — General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette — Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas — Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid — A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem — Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen — Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas — Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas — Towards implementing RL

Other prepublications:
X-88-01 Marc Jumelet — On Solovay's Completeness Theorem

1989
Logic, Semantics and Philosophy of Language:
LP-89-01 Johan van Benthem — The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof — Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema — Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem — Language in Action
LP-89-05 Johan van Benthem — Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj — Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing — The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia — Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang — Dependency of Belief in Distributed Systems

Mathematical Logic and Foundations:
ML-89-01 Dick de Jongh, Albert Visser — Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer — Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna — Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna — On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge — Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen — The Axiomatization of Randomness
ML-89-07 Dirk Roorda — Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda — Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$

Computation and Complexity Theory:
CT-89-01 Michiel H.M. Smid — Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas — Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas — On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet — A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree — A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. — Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi — A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet — Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet — On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel — The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas — Towards Functional Classification of Recursive Query Processing

Other Prepublications:
X-89-01 Marianne Kalsbeek — An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers — New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra — Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof — Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke — The Modal Theory of Inequality
X-89-06 Peter van Emde Boas — Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project

1990 SEE INSIDE BACK COVER

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

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

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

THE SCOPE OF NEGATION IN DISCOURSE

TOWARDS A FLEXIBLE DYNAMIC MONTAGUE GRAMMAR

Paul Dekker
Department of Philosophy
University of Amsterdam

Received July 1990

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

Supported by NWO (Netherlands Organization for Scientific Research)

The Scope of Negation in Discourse*

towards a flexible dynamic Montague grammar

Paul Dekker
ITLI/Department of Philosophy
University of Amsterdam

Abstract

This paper elaborates upon the system of dynamic Montague grammar developed by Groenendijk and Stokhof [11]. This system of natural language interpretation is characterized by the use of dynamic intensional logic, and by a structural raising of the type of sentence translations. We argue that a balanced system of dynamic interpretation requires a higher raising of the type of sentence translations. Next, we show that the required higher order dynamic type assignment is derivable from an old fashioned static type assignment if a version of Hendriks' system of type change is used [15], [16]. The resulting flexible dynamic Montague grammar then is shown to apply successfully to known and new examples that exhibit puzzling and complex anaphoric dependencies.

Natural language semantics in the eighties exhibited an increased interest in the partiality of information and the dynamics of interpretation. This increased interest in both topics has led to the proposal of theories radically departing from Montague's paradigm of natural language interpretation [23], [24]. Notably, situation semantics (Barwise and Perry [3]) and discourse representation theories (Heim [12] and [13], Kamp [20] and Seuren [29]) were proposed as an alternative to classical Montague grammar. However, both kinds of alternatives appeared to be consistent with Montague's program. A version of situation semantics is restated as a partial Montague grammar (Muskens [25]), and basic results of discourse representation theories were captured in dynamic Montague grammar [11].

In this paper we build upon the dynamic renaissance of Montague grammar boosted by Groenendijk and Stokhof. In section 1 we shortly review their dynamic Montague grammar (henceforth, *DMG*) and its successful, fully compositional, treatment of the famous donkey sentences which can be said to lie at the heart of the discourse representation theoretic enterprise. We then turn to the system of extended dynamic interpretation that *DMG* naturally gives rise to. With Groenendijk and Stokhof, we agree that their extended dynamics is corroborated by linguistic facts, but we argue that their account is inevitably incomplete. The extended dynamics of *DMG* originates from a notion of dynamic negation, which itself, we argue, must be rejected.

In section 2 we propose a slight, albeit structural, conservative, modification of *DMG* into *DMG(2)*. We show that the modified framework of interpretation as well allows incorporation of extended dynamic interpretation, this time, however, on the

*I would like to thank Jeroen Groenendijk, Herman Hendriks and Martin Stokhof for their stimulating discussions and constructive comments. The research for this paper was supported by the Foundation for Philosophical Research (SWON), which is subsidized by the Netherlands Organization for Scientific Research. Former versions of this paper were presented at the First European Summer School on Natural Language Processing, Logic and Knowledge Representation, June 1989 in Groningen and at the Seventh Amsterdam Colloquium, December 1989 in Amsterdam.

basis of an adequate notion of dynamic negation. The crucial change involves a raising of the type of sentence translations from the type of sets of propositions to the type of generalized quantifiers over propositions. We show that the proposed definition of dynamic negation enables a neat account of the puzzling interplay between negation and anaphoric reference.

In section 3 we next show that *DMG(2)*'s highly typed translations of natural language expressions can be derived from basically typed translations if a version of Hendriks' system of type change is used [15]. We argue that amended interpretation of the type changes is needed when these deal with downward monotonic expressions. The amendment that we propose, then, is a generalization of the definition of dynamic negation that we introduce in section 2. We show that the resulting system of flexible dynamic Montague grammar (*FDMG*) gives a neat account of all kinds of puzzling inter- and extrasentential dependencies. Section 4, finally, reflects upon possible future developments of *FDMG* and upon some remaining problems.

1 Dynamic Montague grammar

The last decade, natural language semantics has shown increased interest in the dynamics of discourse. An important subject has become the semantic relations that obtain at the intersentential level of discourse. Crucial phenomena are intersentential anaphoric relationships between indefinite noun phrases and pronominal anaphors that occur in different sentential clauses, such as those in examples 1, 2 and 3.

(1) A man walks in the park. He whistles.

(2) If a farmer owns a donkey, he beats it.

(3) Every farmer who owns a donkey beats it.

One may take it to belong to common knowledge by now, that these examples are problematic for a Montague grammar that deals with anaphoric relationships by means of quantification rules. In the first place, we find an anaphoric relationship between noun phrases that occur in different sentences. If quantification rules are used, an indefinite noun phrase must be quantified in in a piece of discourse after all coreferential anaphors have occurred. This treatment can hardly be called compositional in an intuitive sense. In the second place, it gives counterintuitive results if we use quantification rules for establishing the anaphoric relationships in 2 and 3. An alternative approach, therefore, is called for.

Already back in 1968, Karttunen [21] called attention to the context change potential of indefinite noun phrases. Indefinites introduce discourse referents in the domain of discourse, and these discourse referents figure as the referents of pronominal anaphors occurring in sentences to be processed later on. In the eighties, discourse representation theories (*DRT*) have been developed by Kamp [20], Heim [12], [13] and Seuren [29], that gave a formal elaboration of this notion of context change potential, an elaboration that also explains the universal impact of the indefinites in 2 and 3. The basic idea in these theories is that meaning is more than can be captured in terms of truth conditions alone. Certain expressions have, aside from their so called *static*, truth-conditional contribution to meaning, a *dynamic* meaning, which influences the interpretation of sentences to be processed later on. The discourse representation theories introduce an independent level of discourse representation in between the levels of syntactic analysis and semantic interpretation. At this intermediary level discourse markers are introduced by indefinite descriptions and coreference is established between pronominal anaphors and such discourse markers. So, in *DRT* the *representation* of a part of text figures as a context of interpretation for successive parts of text.

The major drawback of *DRT* is that its basic architecture is not compatible with classical Montague grammar, for the most part because of its credited non-compositionality. In standard formulations of *DRT*, the interpretation of natural language expressions is mediated by the (claimed indispensable) discourse representations that they give rise to. In other words, in *DRT* there is no one-to-one map from expressions and syntactic constructions to semantic interpretations (but, cf. Asher [1], Barwise [2], Rooth [28] and Zeevat [30]). *DRT*, then, does not just provide for a fresh view on natural language phenomena, but it implies a complete rebuilding of natural language semantics as well. However, Groenendijk and Stokhof [9], [10], [11] have shown that a compositional reformulation of *DRT* in the style of a Montague grammar is possible. The motivating idea is to stick to the compositional format, and to incorporate the dynamic aspects of meaning right into the meanings associated with basic expressions and construction rules.

Groenendijk and Stokhof developed dynamic predicate logic (*DPL*) as an alternative to *DRT*'s language of discourse representations. The meaning of a sentence in *DPL* is not a set of verifying assignments, but a set of pairs of assignments that form the context change potential of the sentence. The interpretation of a sentence constitutes possible input–output pairs that are used in the sequential processing of discourse. This formalizes the idea that the processing of a sentence brings the interpreter from a certain state of information to another one. Furthermore, by assigning the sentences themselves a dynamic interpretation, it is possible to give a compositional account of interpretation at the intersentential level.

Still, in the construction of a dynamic Montague grammar, the development of *DPL* could only be a first step. *DPL* is just a first order logic and, therefore, it can not be used to give a compositional account of the semantics of natural language at the subsentential level. So, the logical next step was to define a typed logic that encompasses dynamic interpretation. In [11] Groenendijk and Stokhof define a dynamic Montague grammar, based on a typed dynamic logic: dynamic intensional logic (*DIL*). They show that this Montague grammar incorporates basic *DRT*-results in a completely compositional way. Furthermore, they argue that the compositionalization of *DRT* is easily extended to a system that accounts for the dynamics of other quantifying noun phrases than indefinite descriptions only. This extended system forms the starting point of the present paper. It is sketched in section 1.3. Before we come to the extended dynamic system, however, we first show how *DMG* is constructed (section 1.2) on the basis of *DIL* (section 1.1).

1.1 Dynamic intensional logic

In this section we introduce the system of dynamic intensional logic (henceforth, *DIL*), which is used as the framework of interpretation throughout the rest of this paper. *DIL* (Janssen [17], Groenendijk and Stokhof [11]) replaces intensional logic (*IL*) as the language of translation in the dynamic Montague grammars defined below. Using *DIL* we can give a compositional account of dynamic interpretation in natural language, because *DIL* provides for the possibility of dynamic binding. We will speak of dynamic binding when a quantifier binds, semantically, a pronoun translation that is not in the syntactic scope of the quantifier. Basically, this is achieved in *DIL* in the following way.

The language of *DIL* incorporates a distinguished set of discourse markers. These discourse markers act as variables, that is, they figure as pronoun translations, and they are bound by the existential and universal quantifier. The interpretation of the discourse markers is also like that of variables. However, instead of variable assignments, separate discourse markers assignments, which are called *states*, are used to assign a value to them. Likewise, quantifying formulas are interpreted as quantifying over the values of discourse markers instead of over the values of variables. So far, *DIL* does not really differ from *IL*. The crucial difference between

DIL and *IL* is the interpretation of the intension- and extension operators. These operators express abstraction over, and application to states, respectively, that is, abstraction over and application to discourse marker assignments.

Therefore, in *DIL* the intension of a formula is not the set of worlds in which the formula is true, but the set of (discourse marker) assignments with respect to which it is true. This has two important consequences. First, we can take the intension of an expression containing a free discourse marker occurrence and λ-convert it into a context that quantifies over possible values of that discourse marker. As we will see below, this property of *DIL* underlies the possibility of dynamic binding. The second consequence is that, since the intension of a formula in *DIL* is a set of states, instead of worlds, all of *IL*'s intensionality is lost. A proposition in *DIL* contains information about the values of discourse markers, and not information about the world. Wordly information, of course, can be incorporated in *DIL* in a straightforward way (see, for instance, Chierchia [7]). However, for the purposes of this paper, the purely dynamic intensionality of *DIL* will do. (In order to appreciate this point, it may be useful to interpret 'dynamic intensional logic' not as the name of an intensional logic that is dynamic, but as the name of a logic that is dynamically intensional.)

One more thing needs to be noticed before we turn to the definitions of *DIL*. For a first start, and to avoid unnecessary complications, we have discourse markers of type e only. This suffices for a treatment of the whole array of discourse anaphora presented below. An extension of *DIL*'s language with other types of discourse markers is possible (see, for instance, Janssen [17], [18]), but this falls beyond the scope of the present paper. Finally, we have slightly modified *DIL* for ease of exposition. Those who are acquainted with the paper of Groenendijk and Stokhof will notice that we have dropped the state switcher from the language. For as far as the state switcher plays a role in the fragments defined below, its semantics is incorporated in the interpretation of the quantifiers. We now turn to the definition of *DIL*:

The system of dynamic intensional logic is a variant of the system of intensional type theory *IL*, which is used in Montague's *PTQ* [24]. (For further details, see also Janssen [17].) The types of *DIL* are the same as those of *IL*:

Definition 1.1 (Types) T, the set of types, is the smallest set such that:

1. $e, t \in T$
2. If $a, b \in T$, then $\langle a, b\rangle \in T$
3. If $a \in T$, then $\langle s, a\rangle \in T$

As usual, the syntax takes the form of a definition of ME_a, the set of meaningful expressions of type a. Given sets of constants, CON_a, and variables, VAR_a, for every type a, and a set of discourse markers DM, the definition runs as follows:

Definition 1.2 (Syntax) The syntax of *DIL* is like that of *IL* except for the following clauses:

1. If $\alpha \in DM$, then $\alpha \in ME_e$
2. If $\phi \in ME_t, d \in DM$, then $\exists d\phi, \forall d\phi \in ME_t$

The ordinary intensional operators $\Box$ and $\Diamond$ are omitted. The $^\wedge$ and $^\vee$-operators are present, and will be seen to express abstraction over, and application to states (discourse marker assignments) respectively. The quantifiers in (our version of) *DIL* quantify over the values of discourse markers, not over the values of variables.

However, quantification over the values of variables can be defined using the λ-operator, which abstracts over the values of variables in *DIL*.

We now turn to the semantics. Starting from a set D of individuals, and a set S of states, D_a, the domain corresponding to type a, is defined in the familiar fashion:

Definition 1.3 (Domains)

1. $D_e = D$
2. $D_t = \{0, 1\}$
3. $D_{\langle a,b \rangle} = D_b^{D_a}$
4. $D_{\langle s,a \rangle} = D_a^S$

A model M of *DIL* is a pair $\langle D, F \rangle$, where D is as above, and F is a function which interprets the constants of the language. Specifically, if $\alpha \in CON_a$, then $F(\alpha) \in D_a$. Groenendijk and Stokhof also introduce the set of states in the definition of a model, and force these states, by postulate, to behave like discourse marker assignments. However, for matters of conveniency, we will here *identify* the states with discourse marker assignments and, thus, can keep them out of the model. So we have that S, the set of states, or discourse marker assignments, is the set of all functions s such that if $d \in DM$, $s(d) \in D$, and, likewise, G, the set of variable assignments, is the set of all functions g such that if $\nu \in VAR_a$, $g(\nu) \in D_a$.

Now we state the semantics by defining the notion $[\![\alpha]\!]_{M,s,g}$, the interpretation of α with respect to M, s, and g, as follows:

Definition 1.4 (Semantics)

1. $[\![c]\!]_{M,s,g} = F(c)$, for every constant c
   $[\![\nu]\!]_{M,s,g} = g(\nu)$, for every variable ν
   $[\![d]\!]_{M,s,g} = s(d)$, for every discourse marker d
2. $[\![\alpha(\beta)]\!]_{M,s,g} = [\![\alpha]\!]_{M,s,g}([\![\beta]\!]_{M,s,g})$
3. $[\![\neg\phi]\!]_{M,s,g} = 1$ iff $[\![\phi]\!]_{M,s,g} = 0$
   $[\![\phi \wedge \psi]\!]_{M,s,g} = 1$ iff $[\![\phi]\!]_{M,s,g} = [\![\psi]\!]_{M,s,g} = 1$
4. $[\![\exists d\phi]\!]_{M,s,g} = 1$ iff there is a $\mathbf{d} \in D_e$ such that $[\![\phi]\!]_{M,s[d/\mathbf{d}],g} = 1$
   $[\![\forall d\phi]\!]_{M,s,g} = 1$ iff for all $\mathbf{d} \in D_e$ it holds that $[\![\phi]\!]_{M,s[d/\mathbf{d}],g} = 1$
5. $[\![\alpha = \beta]\!]_{M,s,g} = 1$ iff $[\![\alpha]\!]_{M,s,g} = [\![\beta]\!]_{M,s,g}$
6. $[\![\lambda\nu\alpha]\!]_{M,s,g} =$ that function $h \in D_a^{D_b}$ such that $h(\mathbf{d}) = [\![\alpha]\!]_{M,s,g[\nu/\mathbf{d}]}$ for all $\mathbf{d} \in D_b$, where a is the type of α, and b the type of ν
7. $[\![{}^\wedge\alpha]\!]_{M,s,g} =$ that function $h \in D_a^S$ such that $h(s') = [\![\alpha]\!]_{M,s',g}$ for all $s' \in S$, where a is the type of α
8. $[\![{}^\vee\alpha]\!]_{M,s,g} = [\![\alpha]\!]_{M,s,g}(s)$

The notions of truth, validity, entailment and equivalence are defined in the usual way: ϕ *is true with respect to* M, s *and* g iff $[\![\phi]\!]_{M,s,g} = 1$; ϕ is *valid* iff ϕ is true with respect to all M, s, and g; ϕ *entails* ψ iff for all M, s, and g: if $[\![\phi]\!]_{M,s,g} = 1$ then $[\![\psi]\!]_{M,s,g} = 1$; α and β are *equivalent* iff in all M, s, and g: $[\![\alpha]\!]_{M,s,g} = [\![\beta]\!]_{M,s,g}$.

The clauses of definition 1.4 are completely standard, except that in the interpretation of discourse markers, and of formulas that quantify by means of discourse markers, the state parameter is involved, instead of the parameter that assigns values to variables. Now, because the intension operator abstracts over these states, and because the extension operator involves application to them, we have the facility of dynamic binding that we sketched above. First observe the following two familiar facts:

Fact 1.1 ($^{\vee\wedge}$-elimination) $^{\vee\wedge}\alpha$ is equivalent with α

Fact 1.2 (λ-conversion) $(\lambda\nu\alpha)(^{\wedge}\beta)$ is equivalent with $[^{\wedge}\beta/\nu]\alpha$ if all free variables in β are free for ν in α

Notice that in fact 1.2 *no* conditions on discourse markers in β need obtain, because, although their interpretation is state dependent, $^{\wedge}\beta$ is intensionally closed. The following example shows a non-trivial instance of this fact:

$$\lambda p_{\langle s,t\rangle}\exists d(\mathtt{man}(d) \wedge {}^{\vee}p)(^{\wedge}\ \mathtt{whistle}(d)) = \exists d(\mathtt{man}(d) \wedge \mathtt{whistle}(d))$$

We see here that the *DIL*-expression $\lambda p\exists d(\mathtt{man}(d) \wedge {}^{\vee}p)$ is dynamic in the sense indicated above. The embedded existential quantifier, indirectly, binds a discourse marker that occurs in an argument expression of the λ-term, that is, a discourse marker occurring *outside* of the syntactic scope of the quantifier.

1.2 A fragment of natural language

We now turn to the construction of a dynamic Montague grammar. As is usual in such a grammar, basic expressions of natural language are translated into expressions of a logical language with which a model-theoretic interpretation is associated. The interpretation of a compound expression is determined by the meanings of the basic constituent expressions, and operations associated with the rules that are used in the construction of the compound expression. Like we said, the translation language is *DIL*. However, the exposition of *DMG*, and its usage, is facilitated by using a *DMG*-language, which is defined in terms of *DIL*. This language is presented first.

In *DMG* we only use a subset of the types of *DIL*. The types are built up from the types e and $\langle\langle s,t\rangle,t\rangle$, instead of e and t. In fact, the type $\langle\langle s,t\rangle,t\rangle$, will take the part of type t. As we will see shortly, *DMG*-formulas are of this type, and the formulas therefore denote sets of propositions, that is, generalized quantifiers over states. Apart from this complex type of formulas, the definition of *DMG*-types is as usual:

Definition 1.5 (DMG-types)
T_D, the set of *DMG*-types, is the smallest set such that:

1. $e, \langle\langle s,t\rangle,t\rangle \in T_D$
2. If $a, b \in T_D$, then $\langle a,b\rangle \in T_D$
3. If $a \in T_D$, then $\langle s,a\rangle \in T_D$

The types of *DMG* stand in a one-to-one correspondence with *DIL*-types. The following definition of the dynamic type shift, which constitutes a bijection, relates to T to T_D:

Definition 1.6 (Dynamic type shift)

1. $^{\uparrow}e = e$; $^{\uparrow}s = s$
2. $^{\uparrow}t = \langle\langle s,t\rangle,t\rangle$
3. $^{\uparrow}\langle a,b\rangle = \langle^{\uparrow}a,^{\uparrow}b\rangle$

DIL-expressions can be turned into *DMG*-expressions by means of an operation $\uparrow$ (uparrow). The interpretation of a *DMG*-expression $\uparrow\phi$ is given by the following simultaneous recursive definition of $\uparrow$, the interpretation of the type shifts into *DMG*-expressions, and $\downarrow$, the interpretation of shifts back into expressions of the corresponding *DIL*-type (we let ϕ stand for *DIL*-expressions, and Φ for *DMG*-expressions):

Definition 1.7 (Uparrow, downarrow)

1. $\uparrow\phi_e = \phi$
2. $\downarrow\Phi_{\uparrow e} = \Phi$
3. $\uparrow\phi_t = \lambda p_{\langle s,t\rangle}\ (\phi \wedge {}^\vee p)$ (p not free in ϕ)
4. $\downarrow\Phi_{\uparrow t} = \Phi({}^\wedge\mathbf{true})$
5. $\uparrow\phi_{\langle a,b\rangle} = \lambda x_{\uparrow a}\ \uparrow\phi(\downarrow x)$
6. $\downarrow\Phi_{\uparrow\langle a,b\rangle} = \lambda x_a\ \downarrow\Phi(\uparrow x)$
7. $\uparrow\phi_{\langle s,a\rangle} = {}^\wedge\!\uparrow{}^\vee\phi$
8. $\downarrow\Phi_{\uparrow\langle s,a\rangle} = {}^\wedge\!\downarrow{}^\vee\Phi$

The crucial clauses in definition 1.7 are the clauses 3 and 4. The lift of a formula ϕ denotes the sets of propositions true in conjunction with ϕ in the state of evaluation. The lowering of a dynamic formula Φ is its application to the necessarily true proposition. ($\mathbf{true}$ is a constant of type t that is assigned the value 1. One may read it as an abbreviation of the formula $x = x$.) This application has no truth-conditional import, but it amounts to a closure of ϕ's dynamic potential. An easy induction proves the following fact:

Fact 1.3 ($\downarrow\uparrow$-elimination) $\downarrow\uparrow\phi = \phi$

Compare this with ${}^\vee{}^\wedge$-elimination. What does not hold in general is that $\uparrow\downarrow\Phi = \Phi$, as similarly ${}^\wedge{}^\vee\alpha = \alpha$ does not hold in general. We refer to $\uparrow\downarrow\Phi$ as the *static closure* of Φ. The static closure of a dynamic formula strips off the formula's dynamic potential.

The building blocks of *DMG*'s language are constants $\uparrow\mathbf{con}$ derived from *DIL*'s constants $\mathbf{con}$, *DIL*'s variables of *D*-types, and *DIL*'s discourse markers. The language contains abstraction and application, restricted to dynamic expressions, and the intension- and extension-operator, also restricted to dynamic expressions. Furthermore, the language contains three dynamic sentential operators. Dynamic conjunction, dynamic existential quantification and negation are defined as follows:

Definition 1.8 (DMG-operators)

1. $\Phi\,;\Psi = \lambda p_{\langle s,t\rangle}\Phi({}^\wedge(\Psi(p)))$, p not free in Φ or Ψ
2. $\mathcal{E}d\Phi = \lambda p_{\langle s,t\rangle}\exists d\Phi(p)$, p not free in Φ
3. $\sim\!\Phi = \uparrow\neg\downarrow\Phi$

An expression $\Phi\,;\Psi$ represents the dynamic conjunction, or sequence, of two sentences. The dynamic conjunction of two sentences amounts to the intensional composition of the functions that are denoted by the two sentences. Therefore, dynamic conjunction is associative, it is a truly sequential notion of conjunction. Likewise, dynamic existential quantification is the intensional composition of the dynamic existential quantifier with the dynamic formula in its scope. The clauses 1.8.1 and 1.8.2, thus, form the basis for an account of intersentential anaphora, witness fact 1.4:

Fact 1.4 (Associativity)

1. $[\Phi\,;\Psi]\,;\Upsilon = \Phi\,;[\Psi\,;\Upsilon]$

2. $[\mathcal{E}d\Phi]\,;\Psi = \mathcal{E}d[\Phi\,;\Psi]$

Another typical fact is the non-commutativity of dynamic conjunction:

Fact 1.5 (Non-commutativity) $[\Phi\,;\Psi] \neq [\Psi\,;\Phi]$

The notion of negation defined in definition 1.8 is static negation. The negation of Φ in fact is the negation of the closure of Φ, raised to the type of dynamic formulas again. This means that the negation of Φ involves two things: plain negation of the truth-conditional content of Φ *and* a discharge of Φ's dynamic potential. Therefore, the law of double negation does not hold. The double negation of Φ does preserve Φ's truth-conditional content, but not Φ's dynamics. The double negation of Φ equals Φ's static closure.

In terms of the notions of negation, conjunction and existential quantification we define the notions of implication, disjunction and universal quantification in the usual way:

Definition 1.9 (Implication, disjunction and universal quantification)

1. $\Phi \Rightarrow \Psi = {\sim}[\Phi\,;{\sim}\Psi]$

2. Φ **or** $\Psi = {\sim}[{\sim}\Phi\,;{\sim}\Psi]$

3. $\mathcal{A}d\Phi = {\sim}\mathcal{E}d{\sim}\Phi$

By simply applying the definitions, the following facts can be proved:

Fact 1.6

1. $\mathcal{E}d\Phi \Rightarrow \Psi = \mathcal{A}d[\Phi \Rightarrow \Psi]$

2. $[[\Phi\,;\Psi] \Rightarrow \Upsilon] = [\Phi \Rightarrow [\Psi \Rightarrow \Upsilon]]$

3. Φ **or** $\Psi = {\sim}\Phi \Rightarrow \Psi$

Equivalence 1.6 is of paramount importance for a compositional interpretation of the donkey-sentences 2 and 3. The static nature of negation also blocks certain standard equivalences involving these notions. For example: $\mathcal{E}d{\sim}\Phi$ is not equivalent to ${\sim}\mathcal{A}d\Phi$ However, as is to be expected, these formulas do have the same truth-conditional content: ${\downarrow}\mathcal{E}d{\sim}\Phi = {\downarrow}{\sim}\mathcal{A}d\Phi$. So, although the dynamic properties of these constants differ, at the truth-conditional level the existential and the universal quantifier are related in the usual way.

The following equivalences enable us to replace dynamic operators by their static counterparts when we determine the truth-conditions of dynamic formulas by means of $\downarrow$:

Fact 1.7

1. ${\downarrow}{\uparrow}\phi = \phi$

2. $({\uparrow}\phi)(\Psi) = {\uparrow}(\phi({\downarrow}\Psi))$

3. ${\downarrow}{\sim}\Phi = \neg{\downarrow}\Phi$

4. ${\downarrow}\mathcal{E}d\Phi = \exists d{\downarrow}\Phi$

5. ${\downarrow}\mathcal{A}d\Phi = \forall d{\downarrow}\Phi$

6. ${\downarrow}[{\uparrow}\phi\,;\Psi] = \phi \wedge {\downarrow}\Psi$

7. ${\downarrow}[{\uparrow}\phi \Rightarrow \Psi] = \phi \rightarrow {\downarrow}\Psi$

8. $\downarrow[\uparrow\phi \text{ or } \Psi] = \phi \vee \downarrow\Psi$

Like we said, the lowering, or closure, of a formula generates its truth conditions. The closure properties of formulas expressed in fact 1.7 show how we can transform the closure of most dynamic *DMG*-formulas into *DIL*-formulas. We then push the closure operator $\downarrow$ over *DMG*-operators, which are replaced by their static counterparts. The closure operator collapses when it confronts a lifted atomic formula. (The reduction rules in 1.7 do not enable us to reduce all *DMG*-expressions. In section 2.3 we give a complete reduction system.)

We now turn to the construction of a dynamic Montague style fragment of natural language. Compared to traditional Montague grammar the following things are changed. First, of all (constituent-)expressions the type is raised in accordance with the dynamic type shift. Second, the constants are replaced by their raised counterparts and sentential operators by their dynamic counterparts. Finally, in the translation of pronouns and quantifying noun phrases we use indexed discourse markers. The function of the indices is to indicate anaphoric relationships among constituents.

The fragment has as its basic categories the usual categories *IV* (intransitive verb phrases), *CN* (common noun phrases), and *S* (for (sequences of) sentences). Derived categories are of the form *A/B*, *A* and *B* any category. Employed in the fragment are *NP* (= *S/IV*, noun phrases), *Det* (= *NP/CN*, determiners), and *TV* (= *IV/NP*, transitive verb phrases). We first define a function f from categories to *DIL*-types:

Definition 1.10 (Category to type assignment)

1. $f(S) = {}^{\uparrow}t$; $f(CN) = f(IV) = {}^{\uparrow}\langle e,t\rangle$
2. $f(A/B) = \langle\langle s, f(B)\rangle, f(A)\rangle$

We now give some examples of translations of basic expressions. In what follows x and y are variables of type e, P and Q variables of type ${}^{\uparrow}\langle s,\langle e,t\rangle\rangle$, and T of type ${}^{\uparrow}\langle s,\langle\langle s,\langle e,t\rangle\rangle,t\rangle\rangle$; $\mathtt{j}$ is a constant of type e, **man** and **walk** are constants of type $\langle e,t\rangle$, and **love** of type $\langle e,\langle e,t\rangle\rangle$; the d_i are discourse markers.

Definition 1.11 (Translations of basic expressions)

1. *man* $\rightsquigarrow \uparrow\mathbf{man} = \lambda x \uparrow\mathbf{man}(x)$
2. *walk* $\rightsquigarrow \uparrow\mathbf{walk} = \lambda x \uparrow\mathbf{walk}(x)$
3. *love* $\rightsquigarrow \lambda T \lambda x\ {}^{\vee}T({}^{\wedge}\lambda y \uparrow\mathbf{love}(y)(x))$
4. $a_i \rightsquigarrow \lambda P \lambda Q \mathcal{E} d_i[{}^{\vee}P(d_i)\,;{}^{\vee}Q(d_i)]$
5. $every_i \rightsquigarrow \lambda P \lambda Q \mathcal{A} d_i[{}^{\vee}P(d_i) \Rightarrow {}^{\vee}Q(d_i)]$
6. $no_i \rightsquigarrow \lambda P \lambda Q {\sim}\mathcal{E} d_i[{}^{\vee}P(d_i)\,;{}^{\vee}Q(d_i)]$
7. $he_i \rightsquigarrow \lambda Q\ {}^{\vee}Q(d_i)$
8. $John_i \rightsquigarrow \lambda Q \mathcal{E} d_i[\uparrow(\mathtt{j} = d_i)\,;{}^{\vee}Q(d_i)]$

The fragment contains the following construction rules. The interpretations associated with the construction rules are like their ordinary counterparts but for the use of T_D-types and the occurrence of dynamic logical constants.

Definition 1.12 (Construction rules)

1. Functional application: $(\beta_{B/A}\ \alpha_A)_B \rightsquigarrow \beta'({}^{\wedge}\alpha')$

2. Sentence sequencing: $\phi\ .\ \psi \rightsquigarrow \phi' ; \psi'$

3. Conditional sentences: *If* ϕ, *then* $\psi \rightsquigarrow \phi' \Rightarrow \psi'$

4. Restrictive relative clauses: $(\alpha_{CN}\ \beta_S)_{CN} \rightsquigarrow \lambda x\ [\alpha'(x) ; \beta']$

5. Sentence negation: *It is not the case that* $\phi \rightsquigarrow {\sim}\phi'$

We put the system of dynamic Montague grammar to work in the interpretation of dynamic natural language expressions. We review three examples discussed in Groenendijk and Stokhof [11].

A MAN WALKS. HE TALKS

The first example which we discuss, is the sequence of sentences A_i *man walks. He*$_i$ *talks*, a slightly simplified version of our example 1. Like we said, the indices indicate intended anaphoric relationship. Functional application and some standard reduction produce the following translation of the first sentence: $\mathcal{E}d_i[\uparrow\mathtt{man}(d_i) ; \uparrow\mathtt{walk}(d_i)]$. The second sentence has as its reduced translation: $\uparrow\mathtt{talk}(d_i)$. The translation rule for sentence sequences tells us that the translation of the entire sequence is:

$$\mathcal{E}d_i[\uparrow\mathtt{man}(d_i) ; \uparrow\mathtt{walk}(d_i)] ; \uparrow\mathtt{talk}(d_i)$$

The truth-conditions of this formula are determined as follows:

$$\begin{gathered}\downarrow[\mathcal{E}d_i[\uparrow\mathtt{man}(d_i) ; \uparrow\mathtt{walk}(d_i)] ; \uparrow\mathtt{talk}(d_i)] \iff \text{(fact 1.4)}\\ \downarrow\mathcal{E}d_i[\uparrow\mathtt{man}(d_i) ;[\uparrow\mathtt{walk}(d_i) ; \uparrow\mathtt{talk}(d_i)]] \iff \text{(fact 1.7)}\\ \exists d_i(\mathtt{man}(d_i) \wedge \mathtt{walk}(d_i) \wedge \mathtt{talk}(d_i))\end{gathered}$$

So, we see here that the pronoun *he*$_i$ is bound by the quantifying noun phrase a_i *man*, even though its translation does not occur in the immediate syntactic scope of the translation of the noun phrase.

IF A MAN WALKS, HE TALKS

As our second example, we choose the donkey-conditional, which consists of the two sentences which constitute our first example. It is a slight simplification of the real donkey sentence 2. Using the translation rule for conditional sentences, we get the following translation of the sentence:

$$\mathcal{E}d_i[\uparrow\mathtt{man}(d_i) ; \uparrow\mathtt{walk}(d_i)] \Rightarrow \uparrow\mathtt{talk}(d_i)$$

The truth-conditions of the conditional are determined as follows:

$$\begin{gathered}\downarrow[\mathcal{E}d_i[\uparrow\mathtt{man}(d_i) ; \uparrow\mathtt{walk}(d_i)] \Rightarrow \uparrow\mathtt{talk}(d_i)] \iff \text{(fact 1.6)}\\ \downarrow\mathcal{A}d_i[[\uparrow\mathtt{man}(d_i) ; \uparrow\mathtt{walk}(d_i)] \Rightarrow \uparrow\mathtt{talk}(d_i)] \iff \text{(fact 1.6)}\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{man}(d_i) \Rightarrow [\uparrow\mathtt{walk}(d_i) \Rightarrow \uparrow\mathtt{talk}(d_i)]] \iff \text{(fact 1.7)}\\ \forall d_i(\mathtt{man}(d_i) \rightarrow (\mathtt{walk}(d_i) \rightarrow \mathtt{talk}(d_i)))\end{gathered}$$

EVERY FARMER WHO OWNS A DONKEY, BEATS IT

Although the previous example has already shown how donkey-type anaphora are dealt with in *DMG*, we shall also show how the classical donkey-sentence *Every farmer who owns a donkey, beats it* (example 3) is derived. We use the following indexing: *Every*$_i$ *farmer who owns* a_j *donkey, beats it*$_j$. The translations of the main constituents of this sentence read as follows:

own a_j *donkey* $\rightsquigarrow \lambda x\mathcal{E}d_j[\uparrow\mathtt{donkey}(d_j) ; \uparrow\mathtt{own}(d_j)(x)]$
farmer who owns a_j *donkey* $\rightsquigarrow \lambda x[\uparrow\mathtt{farmer}(x) ; \mathcal{E}d_j[\uparrow\mathtt{donkey}(d_j) ; \uparrow\mathtt{own}(d_j)(x)]]$
beat it$_j$ $\rightsquigarrow \lambda x\uparrow\mathtt{beat}(d_j)(x)$

Combining the translation of the determiner *every*$_i$ first with the translation of the complex common noun phrase *farmer who owns a*$_j$ *donkey*, and next with the intransitive verb phrase *beat it*$_j$, we arrive at the following translation of the sentence as a whole:

$$\mathcal{A}d_i[[\uparrow\mathtt{farmer}(d_i)\,;\mathcal{E}d_j[\uparrow\mathtt{donkey}(d_j)\,;\uparrow\mathtt{own}(d_j)(d_i)]] \Rightarrow \uparrow\mathtt{beat}(d_j)(d_i)]$$

The sentence then is assigned the following truth-conditions:

$$\begin{gathered}
\downarrow\mathcal{A}d_i[[\uparrow\mathtt{farmer}(d_i)\,;\mathcal{E}d_j[\uparrow\mathtt{donkey}(d_j)\,;\uparrow\mathtt{own}(d_j)(d_i)]] \Rightarrow \uparrow\mathtt{beat}(d_j)(d_i)] \Longleftrightarrow \\
\downarrow\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i) \Rightarrow [\mathcal{E}d_j[\uparrow\mathtt{donkey}(d_j)\,;\uparrow\mathtt{own}(d_j)(d_i)] \Rightarrow \uparrow\mathtt{beat}(d_j)(d_i)]] \Longleftrightarrow \\
\downarrow\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i) \Rightarrow \mathcal{A}d_j[[\uparrow\mathtt{donkey}(d_j)\,;\uparrow\mathtt{own}(d_j)(d_i)] \Rightarrow \uparrow\mathtt{beat}(d_j)(d_i)]] \Longleftrightarrow \\
\downarrow\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i) \Rightarrow \mathcal{A}d_j[\uparrow\mathtt{donkey}(d_j) \Rightarrow [\uparrow\mathtt{own}(d_j)(d_i) \Rightarrow \uparrow\mathtt{beat}(d_j)(d_i)]]] \Longleftrightarrow \\
\forall d_i(\mathtt{farmer}(d_i) \rightarrow \forall d_i(\mathtt{donkey}(d_j) \rightarrow (\mathtt{own}(d_j)(d_i) \rightarrow \mathtt{beat}(d_j)(d_i))))
\end{gathered}$$

(By means of the facts 1.6 (three times), and fact 1.7.)

This example concludes our exposition of the fragment. What we have shown is that the two central phenomena of cross-sentential anaphora and donkey-sentences, can be treated in *DMG* in an adequate and completely compositional way. In fact, all that distinguishes *DMG* from static theories of interpretation, is the associativity of the dynamic existential quantifier with dynamic conjunction, and the non-commutativity of the last. This means that *DMG* is indeed a semantic theory that unifies important insights from *MG* and *DRT*. We return to the general characteristics of *DMG*, and variants that we will discuss presently, at the end of section 2.

1.3 Extending the dynamics

In the last section of [11] Groenendijk and Stokhof point out that the framework of *DMG* allows for a straightforward extension which enables them to cover certain special examples of anaphoric relationships. On the face of it, these examples involve a kind of dynamic implication, dynamic universal quantification, or dynamic disjunction, and in *DMG* a quite acceptable definition of these notions can be given. The possibility of such an extension, then, serves to indicate "$\cdots$ *that, even restricting ourselves to the first-order level of quantification and anaphoric reference,* DMG *is potentially more than just the sum of* MG *and* DRT." ([11], p. 33). With this we agree.

However, we will argue that only extending *DMG* will not make it cover a whole class of examples, which are structurally related to the examples that Groenendijk and Stokhof take in account. In fact, for a neat account of all these examples, also a slight, but structural, modification of *DMG* into is needed. In the next section we present the required modification of *DMG* into *DMG(2)*. Here we first consider the special examples that *DMG* can handle and the way in which *DMG* accounts for them. Next we argue that Groenendijk and Stokhof's extension of *DMG* must remain incomplete.

Groenendijk and Stokhof discuss the following examples:

(4) It is not the case that John does not own a car. It is red and it is parked in front of the house.

(5) John owns a car. It is red and it is parked in front of the house.

(6) If a client comes in, you treat him politely. You offer him a cup of coffee.

(7) Every player chooses a pawn. He puts it on square one.

(8) Either there is no bathroom here, or it is in a funny place. In any case, it is not on the ground floor.

(9) If there is a bathroom here, it is in a funny place. In any case, it is not on the ground floor.

In example 4 we find a double negation of the sentence *John owns a car*. As appears from the continuation *It is red etc.*, the double negation should not only preserve the truth-conditional content of *John owns a car*, but its dynamic potential as well. The pronoun *it*, seems to be bound by the quantifying noun phrase *a car*, even though this noun phrase is in the scope of the double negation. So, the most likely reading of 4 is equivalent to that of 5, without the double negation. But, as we already said, the notion of negation defined above does not license the law of double negation. The double negation of a formula amounts to the formula's static closure.

Example 6 exhibits a dynamic implication. In this example, the quantifying noun phrase *a client* in the antecedent of the implication not only binds a pronoun in the consequent, with universal force, but also a pronoun in the sentence that follows the implication, again, with universal force. However, the notion of implication defined above licenses the first kind of binding only. The static character of the negation in terms of which the implication is defined blocks possible anaphoric relationships between noun phrases in an implication and pronouns that follow it. The same goes for example 7. The negation in terms of which the universal quantifier is defined blocks dynamic binding by the noun phrases *every player* and *a pawn*. Still, the pronouns *he* and *it* seem to be bound by these two noun phrases. Example 8 exhibits two kinds of binding which we cannot account for as yet. In the first place, the noun phrase *no bathroom* in the first disjunct cannot bind the pronoun *it* in the second disjunct since the determiner *no* is static and since the (second) negation in the definition of the disjunction blocks binding of pronouns in the second disjunct by noun phrases in the first. Furthermore, the negations in the definition of the disjunction block anaphoric relationships between noun phrases figuring in the disjunction and pronouns following it, so the pronoun in the second sentence in 8 remains unbound as well. Now, if the classical equivalence between $\Phi \Rightarrow \Psi$ and $\sim\Phi$ **or** Ψ is valid (which it is not at present), and if the law of double negation holds unconditionally, we can account for the first kind of binding. Furthermore, if the implication is dynamic (as required for a proper account of example 6), then also the second kind of binding is licensed. Example 9, which we can take to be equivalent to example 8, may serve to illustrate the need to validate these classical equivalences also at the dynamic level.

The nice thing about *DMG* is that it is easily modified into a system that accounts for the anaphoric relationships in 4–9, and derives plausible readings for these examples. What is more, in order to have this we only need to redefine negation in a trivial way. Consider the following, standard, definition of negation as complementation:

Definition 1.13 (Dynamic negation) $\sim\Phi = \lambda p\ \neg\Phi(p)$

The old definition 1.8.3 of static negation can be obtained from the dynamic one, by taking the static closure of the dynamic negation: $\uparrow\downarrow\sim\Phi = \uparrow\neg\downarrow\Phi$ Now, if we adopt the notion of negation as defined in 1.13, the law of double negation holds, which enables an account of the equivalence of the examples 4 and 5:

Fact 1.8 $\sim\sim\Phi = \Phi$

The other dynamic operators are defined in the usual way, this time in terms of the notion of dynamic negation:

Definition 1.14 (Implication, disjunction and universal quantification)

1. $\Phi \Rightarrow \Psi = \sim[\Phi\,;\sim\Psi]$

2. Φ **or** $\Psi = \sim[\sim\Phi\,;\sim\Psi]$

3. $\mathcal{A}d\Phi = \sim\mathcal{E}d\sim\Phi$

Although these notions are defined with the help of dynamic negation, the equivalences in fact 1.6 still hold and fact 1.4 and the reduction rules in 1.7 remain valid as well. However, since we use the dynamic notion of negation from 1.13, we recover the full interdefinability of the quantifiers, and of $\Rightarrow$ and **or**:

Fact 1.9

1. $\mathcal{E}d\sim\Phi = \sim\mathcal{A}d\Phi$

2. $\Phi \Rightarrow \Psi = \sim\Phi$ **or** Ψ

What is more important, we now also have associativity of $\Rightarrow$, **or** and $\mathcal{A}d$ with dynamic conjunction:

Fact 1.10 (Extended associativity)

1. $[\Phi \Rightarrow \Psi]\,;\Upsilon = \Phi \Rightarrow [\Psi\,;\Upsilon]$

2. $[\Phi$ **or** $\Psi]\,;\Upsilon = \Phi$ **or** $[\Psi\,;\Upsilon]$

3. $[\mathcal{A}d\Phi]\,;\Psi = \mathcal{A}d[\Phi\,;\Psi]$

This fact expresses the dynamic properties of the dynamic operators. These dynamic properties enable a treatment of the examples 6–9. We present the treatment of (slight variants of) examples 6–8, the treatment of 9 runs parallel to that of 6.

IF A CLIENT COMES IN, YOU PAMPER HIM. YOU OFFER HIM COFFEE
Functional application and some reductions give the following translation of 6:

$$[\mathcal{E}d_i[\uparrow\mathtt{client}(d_i)\,;\uparrow\mathtt{come}(d_i)] \Rightarrow \uparrow\mathtt{pamper}(d_i)(\mathtt{y})]\,;\uparrow\mathtt{offer}(\mathtt{c})(d_i)(\mathtt{y})$$

This expression is assigned the following truth-conditions:

$$\begin{gathered}\downarrow[[\mathcal{E}d_i[\uparrow\mathtt{client}(d_i)\,;\uparrow\mathtt{come}(d_i)] \Rightarrow \uparrow\mathtt{pamper}(d_i)(\mathtt{y})]\,;\uparrow\mathtt{offer}(\mathtt{c})(d_i)(\mathtt{y})] \Longleftrightarrow\\ \downarrow[\mathcal{E}d_i[\uparrow\mathtt{client}(d_i)\,;\uparrow\mathtt{come}(d_i)] \Rightarrow [\uparrow\mathtt{pamper}(d_i)(\mathtt{y})\,;\uparrow\mathtt{offer}(\mathtt{c})(d_i)(\mathtt{y})]] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[[\uparrow\mathtt{client}(d_i)\,;\uparrow\mathtt{come}(d_i)] \Rightarrow [\uparrow\mathtt{pamper}(d_i)(\mathtt{y})\,;\uparrow\mathtt{offer}(\mathtt{c})(d_i)(\mathtt{y})]] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{client}(d_i) \Rightarrow [\uparrow\mathtt{come}(d_i) \Rightarrow [\uparrow\mathtt{pamper}(d_i)(\mathtt{y})\,;\uparrow\mathtt{offer}(\mathtt{c})(d_i)(\mathtt{y})]]] \Longleftrightarrow\\ \forall d_i(\mathtt{client}(d_i) \rightarrow (\mathtt{come}(d_i) \rightarrow (\mathtt{pamper}(d_i)(\mathtt{y}) \wedge \mathtt{offer}(\mathtt{c})(d_i)(\mathtt{y}))))\end{gathered}$$

(By means of facts 1.10, 1.6 (twice) and 1.7 respectively.)

EVERY PLAYER CHOOSES A PAWN. HE PUTS IT ON SQUARE ONE
Functional application and some reductions give the following translation of 7:

$$\mathcal{A}d_i[\uparrow\mathtt{player}(d_i) \Rightarrow \mathcal{E}d_j[\uparrow\mathtt{pawn}(d_j)\,;\uparrow\mathtt{choose}(d_j)(d_i)]]\,;\uparrow\mathtt{put\_on}(\mathtt{1})(d_j)(d_i)$$

This expression is assigned the following truth-conditions:

$$\begin{gathered}\downarrow[\mathcal{A}d_i[\uparrow\mathtt{player}(d_i) \Rightarrow \mathcal{E}d_j[\uparrow\mathtt{pawn}(d_j)\,;\uparrow\mathtt{choose}(d_j)(d_i)]]\,;\uparrow\mathtt{put\_on}(\mathtt{1})(d_j)(d_i)] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{player}(d_i) \Rightarrow \mathcal{E}d_j[\uparrow\mathtt{pawn}(d_j)\,;[\uparrow\mathtt{choose}(d_j)(d_i)\,;\uparrow\mathtt{put\_on}(\mathtt{1})(d_j)(d_i)]]] \Longleftrightarrow\\ \forall d_i(\mathtt{player}(d_i) \rightarrow \exists d_j(\mathtt{pawn}(d_j) \wedge \mathtt{choose}(d_j)(d_i) \wedge \mathtt{put\_on}(\mathtt{1})(d_j)(d_i)))\end{gathered}$$

(By means of the facts 1.10 and 1.4 and fact 1.7 respectively.)

There is no bathroom here or it is downstairs. It is not upstairs
Functional application and some reductions give the following translation of 8:

$$[\sim\mathcal{E}d[\uparrow\mathtt{bathroom}(d)\,;\uparrow\mathtt{here}(d)]\ \mathbf{or}\ \uparrow\mathtt{down}(d)]\,;\sim\uparrow\mathtt{up}(d)$$

This expression is assigned the following truth-conditions:

$$\begin{array}{c}\downarrow[[\sim\mathcal{E}d[\uparrow\mathtt{bathroom}(d)\,;\uparrow\mathtt{here}(d)]\ \mathbf{or}\ \uparrow\mathtt{down}(d)]\,;\sim\uparrow\mathtt{up}(d)]\iff\\ \downarrow[\sim\mathcal{E}d[\uparrow\mathtt{bathroom}(d)\,;\uparrow\mathtt{here}(d)]\ \mathbf{or}\ [\uparrow\mathtt{down}(d)\,;\sim\uparrow\mathtt{up}(d)]]\iff\\ \downarrow[\mathcal{E}d[\uparrow\mathtt{bathroom}(d)\,;\uparrow\mathtt{here}(d)]\Rightarrow[\uparrow\mathtt{down}(d)\,;\sim\uparrow\mathtt{up}(d)]]\iff\\ \downarrow\mathcal{A}d[[\uparrow\mathtt{bathroom}(d)\,;\uparrow\mathtt{here}(d)]\Rightarrow[\uparrow\mathtt{down}(d)\,;\sim\uparrow\mathtt{up}(d)]]\iff\\ \forall d((\mathtt{bathroom}(d)\wedge\mathtt{here}(d))\rightarrow(\mathtt{down}(d)\wedge\neg\mathtt{up}(d)))\end{array}$$

(By means of the facts 1.10, 1.9, 1.6 and 1.7 respectively.)

So far, it seems there is only reason to cheer. Just by reformulating *MG* in order to capture *DRT*-results in a compositional way, we get a system that naturally accounts for phenomena which it wasn't designed to account for in the first place. However, the extended dynamics is based, crucially, on a feature of dynamic negation which Groenendijk and Stokhof point at and which is not pleasant. Dynamic negation is associative with conjunction:

Fact 1.11 $\sim\Phi\,;\Psi=\sim[\Phi\,;\Psi]$

Obviously, this is something that one does not want. Fact 1.11 says that the negation of a sentence S extends to sentences that follow S in the discourse. So, whereas the static negation of Φ involves negation of truth-conditional content, but does not preserve Φ's dynamic potential, this dynamic negation of Φ does preserve Φ's dynamics, but it does not involve negation of just the truth-conditional content of Φ. The dynamic ngeation involves negation of the content of Φ conjoined with all sentences that follow Φ. We can accurately restate this problem in terms of the monotonicity of *DMG*-formulas. In the original version of *DMG* all sentences denote upward monotonic quantifiers over states. This conforms to intuition, because it means that if sentence T is more informative than sentence U, then the result of sequencing a sentence S with T is more informative than the result of sequencing S with U. Furthermore, upward monotonicity guarantees that (non trivial) sentence sequencing always involves information update, because for any two dynamic formulas Φ and Ψ, if Φ is upward monotonic, then $\downarrow[\Phi\,;\Psi]$ entails Φ. However, with the notion of dynamic negation at hand, things get different. The dynamic negation of an upward monotonic formula returns a downward monotonic formula, and if Φ is a downward monotonic formula, then sequencing Φ amounts to information 'downdate':

$$\downarrow\Phi \text{ entails } \downarrow[\Phi\,;\Psi] \text{ if } \Phi \text{ is downward monotonic}$$

Groenendijk and Stokhof propose to secure upward monotonicity and information update in *DMG* by imposing constraints on translations. The translation of each sentence which constitutes a separate step in the discourse may not be downward monotonic. As Groenendijk and Stokhof remark, this constraint must not be imposed on all sentence translations, because then, of course, the crucial dynamic negation itself could never be used. The constraint should only apply to certain constructions. It must be invoked in discourse sequencing, in relative clause formation and in the formation of conditional sentences.

Apart from a more general feeling of unease about constraints, the constraints under consideration itch. For instance, things get troublesome when the fragment is extended with the dynamic negation of other than sentential expressions. Groenendijk and Stokhof themselves propose a dynamic interpretation of the determiner *no* which guarantees upward monotonicity:

$$\lambda P \lambda Q \sim \mathcal{E} d[{}^{\vee}P(d)\,;\sim\uparrow\downarrow\sim{}^{\vee}Q(d)]$$

This enables them to account for the following examples:

(10) No player leaves the room. He stays were he is.

(11) No client that comes in is offered coffee. He is directly sent up to me.

However, this upward monotonic interpretation of the determiner *no* undermines Groenendijk and Stokhof's own approach to the bathroom disjunction (8) *Either there is no bathroom here or it is in a funny place. In any case, it is not on the ground floor.* If the first disjunct *there is no bathroom here* in the dynamic disjunction is allowed to be upward monotonic, then 8 comes to mean that there is no bathroom here, and that every bathroom that is in a funny place is on the ground floor. Of course, the constraint that the antecedent of a conditional be upward monotonic, should be mirrored by a constraint that the first disjunct of dynamic disjunction be downward monotonic. But then, in order to be able to interpret 8, we need a downward monotonic interpretation of *no* again, that is, we must require it to be ambiguous.

Furthermore, the proposed dynamic interpretation of *no* that guarantees upward monotonicity, closes off the dynamic potential of the second argument of *no*. This should not be, witness the following example:

(12) No farmer beats a donkey he owns. He doesn't kick it either.

If we use the upward monotonic interpretation of *no* in 12, the pronoun *it* can not be bound by the (closed) quantifying noun phrase *a donkey he owns.* On the other hand, if we use the downward monotonic interpretation, 12 comes out to mean that for every farmer and every donkey that he owns it holds that if the farmer beats the donkey, he also kicks it. A plausible interpretation would result if we could take *No farmer beats a donkey he owns* to be equivalent with *Every farmer does not beat every donkey he owns,* with a static negation of the *TV beat.* However, only ad hoc amendations will give us this.

We can also find examples of dynamic implication and dynamic disjunction which Groenendijk and Stokhof's proposal excludes. For instance, in example 13 we have a dynamic disjunction with an upward monotonic first disjunct, and in the examples 14 and 15 we find an existential quantifier that is in the scope of a negation in the antecedent, but that nevertheless binds a pronoun in the consequent:

(13) Either there is a bathroom downstairs, or it is upstairs.

(14) If it is not the case that there is a bathroom downstairs, then it is upstairs.

(15) If a chessbox doesn't contain a spare pawn, then it is taped on top of it.

If we forget the constraints for a minute, examples 13 and 14 turn out to mean that there is a bathroom that is downstairs and not upstairs. Example 15 can be read in two different ways. We can take the negation to be *TV*-negation, and the interpretation that results then is that every chessbox contains every spare pawn and that no spare pawn is on top of any chessbox. We can as well take the negation to be *IV*-negation, and derive the interpretation that every chessbox contains a spare pawn that is not on top of it. In any case the interpretations of 13–15 are wrong. If we follow Groenendijk and Stokhof, all we can do about these examples is exclude them, on behalf of the monotonicity constraint, but this is hardly acceptable.

So, we think, Groenendijk and Stokhof's constraints miss the point. It seems that the notion of dynamic negation is in need of revision. Now let us consider what an adequate definition of dynamic negation should be like. We can think of two requirements. First, of course, we want the law of double dynamic negation to hold. The validity of this law ensures that dynamic negation preserves dynamic potential, which is what we need in order to cope with the examples 4–15. Second, we want that $\sim\uparrow\phi = \uparrow\neg\phi$. This requirement must ensure that dynamic negation involves negation of truth-conditional content. Clearly, since $\uparrow\phi$ has no dynamic potential whatsoever, its dynamic negation $\sim\uparrow\phi$ should amount to the negation of the truth-conditions of ϕ, and nothing more. Furthermore, once we have that $\sim\uparrow\phi = \uparrow\neg\phi$, a simple induction shows that all dynamic formulas constructed from atomic formulas with the help of $\uparrow$, $\mathcal{E}$, $\mathcal{A}$, ;, $\Rightarrow$, **or** and $\sim$ are upward monotonic. (As a corollary, this monotonicity naturally refutes the associativity of dynamic negation with dynamic conjunction, which is expressed in fact 1.11. Like we said, this associativity should be ruled out in the first place.)

So, we have two requirements that a notion of dynamic negation must satisfy. We know that negation in the restricted version of *DMG* obeys the second requirement, but not the first, and that negation in the extended version of *DMG* obeys the first requirement, but not the second. The natural question then is whether it is possible to define a notion of dynamic negation within the framework of *DMG* that obeys both requirements. And the straighforward answer to that question is that it is impossible, since the only possible definition of negation that gives way to the equivalence $\sim\uparrow\phi = \uparrow\neg\phi$ is static negation. We conclude that no adequate definition of dynamic negation within the framework of *DMG* is possible and that the extended version of *DMG* exhibits a structural gap, precisely because it is based on an inadequate definition of dynamic negation.

Where does this leave us? We have examined *DMG* and the way in which it gives a true compositionalization of *DRT*. Furthermore, with Groenendijk and Stokhof, we enjoyed *DMG*'s inherent possibility to treat extended dynamics of natural language. However, we observed a gap in the extended dynamic system, the origins of which were traced back to the roots of the extension itself: dynamic negation. We next pointed out that it is impossible to give a proper definition of dynamic negation within the framework of *DMG*, and we concluded that *DMG* does not lay the proper foundation for an extended dynamic system of interpretation. So, it seems that the extended dynamic prospect was an illusion after all. However, in the next section we show that a slight, structural, modification of *DMG* into *DMG(2)* makes it possible to come up with a definition of dynamic negation that satisfies the requirements we imposed on it.

2 Quantification over propositions

In this section we show that a notion of dynamic negation can be defined adequately, if we raise the *DMG* type of formulas one more time. In our proposal, labeled *DMG(2)*, formulas denote generalized quantifiers over propositions instead of sets of propositions (generalized quantifiers over states). We first introduce this raised fragment, and then come back to the issue of dynamic negation again.

2.1 Dynamic Montague grammar (2)

The basic difference between *DMG* and *DMG(2)* resides in the type assigned to formulas. If we let τ abbreviate the type $\langle\langle s,t\rangle,t\rangle$ of *DMG*-formulas, then the type of *DMG(2)*-formulas is $\langle\langle s,\tau\rangle,t\rangle$, the type of generalized quantifiers over propositions. The definition of the *DMG(2)*-types runs as follows:

Definition 2.1 (DMG(2) types)
$T_{D(2)}$, the set of *DMG(2)* types, is the smallest set such that:

1. $e, \langle\langle s,\tau\rangle,t\rangle \in T_{D(2)}$

2. If $a, b \in T_{D(2)}$, then $\langle a, b\rangle \in T_{D(2)}$
3. If $a \in T_{D(2)}$, then $\langle s, a\rangle \in T_{D(2)}$

We redefine the type shift from *DIL*-types into *DMG(2)*-types in the following way:

Definition 2.2 (Dynamic type shift(2))

1. ${}^{\uparrow}e = e$; ${}^{\uparrow}s = s$
2. ${}^{\uparrow}t = \langle\langle s, \tau\rangle, t\rangle$
3. ${}^{\uparrow}\langle a, b\rangle = \langle {}^{\uparrow}a, {}^{\uparrow}b\rangle$

Notice that the set of *DMG(2)* types $T_{D(2)}$ again is a subset of the set of *DIL* types T. We also define a type shift $^{\Downarrow}$ from $T_{D(2)}$ to T_D, which relates the *DMG(2)* types to the corresponding *DMG* types:

Definition 2.3 (Type shift from $T_{D(2)}$ to T_D)

1. ${}^{\Downarrow\uparrow}e = e$; ${}^{\Downarrow\uparrow}s = s$
2. ${}^{\Downarrow\uparrow}t = \langle\langle s, t\rangle, t\rangle$
3. ${}^{\Downarrow\uparrow}\langle a, b\rangle = \langle {}^{\Downarrow\uparrow}a, {}^{\Downarrow\uparrow}b\rangle$

DIL-expressions will now be turned into *DMG(2)*-expressions by means of a redefined operation $\Uparrow$. The interpretation of a *DMG(2)*-expression $\Uparrow\phi$ is given by the following simultaneous recursive definition of $\Uparrow$, the interpretation of type shifts into *DMG(2)*-expressions, and $\Downarrow$, the interpretation of shifts back into expressions of the corresponding *DIL*-type:

Definition 2.4 (Uparrow(2), downarrow(2))

1. $\Uparrow\phi_e = \phi$
2. $\Downarrow\Phi_{\uparrow e} = \Phi$
3. $\Uparrow\phi_t = \lambda R_{\langle s,\tau\rangle}{}^{\vee}R({}^{\wedge}\phi)$ (R not free in ϕ)
4. $\Downarrow\Phi_{\uparrow t} = \Phi({}^{\wedge}\lambda p_{\langle s,t\rangle}{}^{\vee}p)$
5. $\Uparrow\phi_{\langle a,b\rangle} = \lambda x_{\uparrow a}\ \Uparrow\phi(\Downarrow x)$
6. $\Downarrow\Phi_{\uparrow\langle a,b\rangle} = \lambda x_a\ \Downarrow\Phi(\Uparrow x)$
7. $\Uparrow\phi_{\langle s,a\rangle} = {}^{\wedge}\Uparrow{}^{\vee}\phi$
8. $\Downarrow\Phi_{\uparrow\langle s,a\rangle} = {}^{\wedge}\Downarrow{}^{\vee}\Phi$

The crucial clauses in definition 2.4 are clause 3 and 4 again. The raising of a formula ϕ denotes the properties of the proposition *that* ϕ. The lowering of a dynamic formula Φ comes down to the statement that Φ has the property of being a true proposition. Again, the lowering of a dynamic formula Φ gives us Φ's truth-conditions, with its dynamics stripped of. The following fact remains valid:

Fact 2.1 ($\Downarrow\Uparrow$-elimination(2)) $\Downarrow\Uparrow\phi = \phi$

What still does not hold in general is that $\Uparrow\Downarrow\Phi = \Phi$. The operator $\Downarrow$ functions as a kind of closure operator, one which closes off a (piece of) text.

The language of *DMG(2)* is constructed in complete analogy to the language of *DMG*. The constants $\Uparrow$**con** of *DMG(2)* are derived from the constants **con** of *DIL*, *DMG(2)*'s discourse markers are *DIL*'s discourse markers, and *DMG(2)*'s variables are *DIL*'s variables of the raised dynamic type. *DMG(2)*'s language contains abstraction and application of dynamic expressions, and the intension- and extension-operator, restricted to dynamic expressions. Only the interpretation of dynamic conjunction is changed, and dynamic existential quantification and static negation differ with repect to the associated types:

Definition 2.5 (DMG-operators(2))

1. $\Phi\,;\Psi = \lambda R_{\langle s,\tau\rangle}\Phi(^\wedge\lambda p\ (^\vee p \wedge \Psi(R)))$, R not free in Φ or Ψ
2. $\mathcal{E}d\Phi = \lambda R_{\langle s,\tau\rangle}\exists d\Phi(R)$, R not free in Φ
3. $\sim\!\Phi = \uparrow\neg\downarrow\Phi$

The dynamic conjunction of Φ and Ψ now denotes the set of properties R of propositions such that Φ has the property of being true in conjunction with the fact that R is a property of Ψ. The clauses 2.5.1 and 2.5.2 as well guarantee associativity and non-commutativity:

Fact 2.2 (Associativity(2), non-commutativity(2))

1. $[\Phi\,;\Psi]\,;\Upsilon = \Phi\,;[\Psi\,;\Upsilon]$
2. $[\mathcal{E}d\Phi]\,;\Psi = \mathcal{E}d[\Phi\,;\Psi]$
3. $[\Phi\,;\Psi] \neq [\Psi\,;\Phi]$

The definitions of implication, disjunction and universal quantification are the same as in *DMG*, but they are based on the *DMG(2)* definition of ;, $\mathcal{E}d$, and $\sim$:

Definition 2.6 (Implication disjunction and universal quantification(2))

1. $\Phi \Rightarrow \Psi = \sim[\Phi\,;\sim\Psi]$
2. Φ **or** $\Psi = \sim[\sim\Phi\,;\sim\Psi]$
3. $\mathcal{A}d\Phi = \sim\mathcal{E}d\sim\Phi$

The categories and the syncategorematic constructions of the fragment remain the same. The function f from *DMG(2)*-categories to types differs only with respect to the interpretation of $^\uparrow$:

Definition 2.7 (Category to type assignment(2))

1. $f(S) = {}^\uparrow t$; $f(CN) = f(IV) = {}^\uparrow\langle e,t\rangle$
2. $f(A/B) = \langle\langle s, f(B)\rangle, f(A)\rangle$

The relation with the ordinary *MG*-types remains obvious: t is replaced by $\uparrow t$.

The translations of basic expressions remain the same, except that the *DMG(2)*-interpretation of the abbreviations is used. We will not repeat the fragment here, because it is typographically identical to the fragment defined in 1.11. The only real change is in the types assigned to the expressions, and in the interpretation of the abbreviations. The interpretations of the syncategorematic constructions are typographically identical as well.

We will not show the determination of the truth-conditions of the sentences treated above in the framework of the *DMG*-fragment. Instead, we prove that all *DMG*-sentences are assigned the same truth-conditions in *DMG* and *DMG(2)*. For this, we first define an operation $\Downarrow$ which turns expressions of the *DMG(2)*-type into expressions of the corresponding *DMG*-type (see the definition of the type shift $^\Downarrow$ in 2.3). The interpretation of $\Downarrow\Phi$ is given by the following simultaneous recursive definition of $\Downarrow$ and $\Uparrow$:

Definition 2.8

1. $\Downarrow\Phi_{\uparrow e} = \Phi$
2. $\Uparrow\phi_{\Downarrow\uparrow e} = \phi$
3. $\Downarrow\Phi_{\uparrow t} = \lambda q_{\langle s,t\rangle}\ \Phi({}^\wedge\lambda p\ ({}^\vee p \wedge {}^\vee q))$, R not free in ϕ
4. $\Uparrow\phi_{\Downarrow\uparrow t} = \lambda R_{\langle s,\tau\rangle}\phi({}^\wedge\ {}^\vee R({}^\wedge\mathtt{true}))$, q not free in ϕ
5. $\Downarrow\Phi_{\uparrow\langle a,b\rangle} = \lambda x_{\Downarrow\uparrow a}\ \Downarrow\Phi(\Uparrow x)$
6. $\Uparrow\phi_{\Downarrow\uparrow\langle a,b\rangle} = \lambda x_{\uparrow a}\ \Uparrow\phi(\Downarrow x)$
7. $\Downarrow\Phi_{\uparrow\langle s,a\rangle} = {}^\wedge\Downarrow{}^\vee\Phi$
8. $\Uparrow\phi_{\Downarrow\uparrow\langle s,a\rangle} = {}^\wedge\Uparrow{}^\vee\phi$

In the appendix, we prove the following two facts. Let $\Phi^\dagger$ be the *DIL*-equivalent of a *DMG*-expression Φ, and $\Phi^\ddagger$ the *DIL*-equivalent of the *DMG(2)*-expression Φ, then:

Fact 2.3 If Φ is a *DMG*-expression without free variables, then $\Downarrow\Phi^\ddagger = \Phi^\dagger$

Fact 2.3 shows that the translation $\Downarrow\Phi$ of a *DMG(2)*-expression Φ gives us the *DMG*-interpretation Φ. Next, let $\downarrow^\dagger$ furthermore denote the *DMG* interpretation of $\downarrow$ as defined in 1.7, and $\downarrow^\ddagger$ the *DMG(2)* interpretation of it defined in 2.4, then:

Fact 2.4 If $\Phi^\ddagger$ is a *DMG*-expression without free variables, then $\downarrow^\ddagger\Phi^\ddagger = \downarrow^\dagger\Downarrow\Phi^\ddagger$

The facts 2.3 and 2.4 suffice to prove the equivalence of *DMG(2)* and *DMG*. Since the translations of natural language expressions in *DMG(2)* and *DMG* are syntactically identical and do not contain free variables, these facts show that the truth-conditions of a sentence S in *DMG(2)*: $\downarrow^\ddagger S'^\ddagger = \downarrow^\dagger\Downarrow S'^\ddagger = \downarrow^\dagger S'^\dagger$ are the sentence's truth-conditions in *DMG*.

Those who inspect the appendix observe that the restriction to *DMG*-expressions without free variables is crucial for the validity of the equivalence fact. Those who do not inspect the appendix may take this observation for granted. One question naturally emerges in view of this observation, the question what these free variables could stand for if they were there. There is a clear answer to this: these variables can denote functions f that do not obey the reduction scheme: $\downarrow(f(\alpha)) = (\downarrow f)(\uparrow\alpha)$. In other words, these functions would be inherently dynamically intensional and would not be derivable from extensional (static) functions.

For the present purposes it is reasonable to restrict our attention to dynamic semantic objects that, in the end, derive from static objects. However, one might think of incorporating atoms in the language that are inherently dynamic intensional. A case in point might be the verb *believe*. Treating this verb as a dynamic intensional verb might give way to an interesting new treatment of belief sentences next to two well-known approaches to the semantics of belief sentences. According to one approach, *believe* is an intensional verb that characterizes a subject's disposition to *act*. According to another approach, it is a hyperintensional verb that characterizes a subject's disposition to *assent*. In a dynamic Montague grammar, we can treat *believe* as a dynamic intensional verb, and this may very well characterize a subject's disposition to *comment*, or *react*. However, we will not pursue an analysis of belief along these lines here, but we leave it at this vague hint. Still, notice that we may preserve the equivalence of *DMG* and *DMG(2)* if we introduce such dynamic intensional verbs in the language, that is, if we disregard expressions with free variables again. If we allow such verbs in the language and if we want to preserve the equivalence, then we only need to use as dynamic intensional constants in *DMG(2)* the raising $\Uparrow\mathbf{con}$ of dynamic intensional constants $\mathbf{con}$ in *DMG*.

2.2 Dynamic negation

We saw above that *DMG* is potentially more then *DRT*, and the same holds of course for *DMG(2)*. If we interpret negation as complementation in *DMG(2)*, we get a system provably equivalent to the extended dynamic variant presented in section 1.3. (The translation algorithm $\Downarrow$ maps *DMG(2)* with negation as complementation onto the extended version of *DMG*.) But, of course, in that case we inherit both virtue and vice of that dynamic negation. However, whereas the language of *DMG* leaves no room for an adequate definition of dynamic negation, the language of *DMG(2)* does. The language of *DMG(2)* is more expressive than that of *DMG*, and provides for the logical space necessary for an adequate definition of dynamic negation. In this section we present our definition of dynamic negation, we show that it has the logical properties that we required it to have above, and we show that it applies successfully to cases of dynamic negation in natural language.

In *DMG(2)*, sentences have the same monotonicity properties as in *DMG*. All *DMG(2)* formulas denote upward monotonic quantifiers over propositions. Furthermore, as a direct consequence of fact 2.4, sentence sequencing in *DMG(2)* always produces information update, like it does in *DMG*. Our task is to define a notion of dynamic negation that preserves both upward monotonicity *and* dynamic potential of the formulas negated.

When a dynamic formula is sequenced in *DMG(2)*, its interpretation is applied to the property of propositions *that their conjunction with the information expressed by the continuation of the discourse holds true.* This application comes down to the statement that the interpretation of the first sequent remains true under a certain extension of information. In fact, something similar holds for classical (intensional) conjunction. In classical conjunction, one might say, the property of propositions *to be true in conjunction with the second conjunct* is applied to the proposition expressed by first. In the system of *DMG(2)*, this property of propositions is said to be a property of the quantifier over propositions expressed by the first sequent. Therefore, the crucial difference with classical conjunction is that the function argument structure is turned around. (And, since we use *DIL*, a corollary of this reversed application structure is that the information expressed by further discourse may be in the dynamic scope of expressions in the first sequent.)

In a dynamic Montague grammar we are interested in the properties of propositions that quantifiers over propositions have. When we turn to the dynamic negation of such quantifiers, the focus switches to the properties of propositions that these quantifiers do not have. The similarity with the classical case of conjunction now guides us towards a proper understanding of dynamic negation. Clearly, if in the classical case the negation of ϕ is conjoined with ψ, then the result does not mean that ϕ has the property of propositions whose truth is not preserved when extended with the information expressed by ψ. This wouldn't entail anything about the truth or falsity of ϕ, nor of ψ. Likewise, it is wrong to define dynamic negation in *DMG(2)* as complementation. If we did so, then a sequence with a negated dynamic formula Φ as the first sequent amounts to the statement that the truth of Φ is not preserved under extension with the contents of the continuation of discourse. However, what the classical conjunction does attribute of the proposition expressed by ϕ when its negation is conjoined with ψ, is that it does not preserve truth under a weakening of information expressed by the disjunction with $\neg\psi$, witness the following equivalence:

$$(\lambda p\ (\neg{}^{\vee}p \wedge \psi))({}^{\wedge}\phi) = \neg(\lambda p\ ({}^{\vee}p \vee \neg\psi))({}^{\wedge}\phi)$$

In other words, when the negation of ϕ is classically conjoined with ψ, then it is this property of propositions $\lambda p\ {}^{\vee}p \vee \neg\psi$, that is said not to apply to ϕ. The same kind of reasoning holds at the dynamic level of *DMG(2)*. When in *DMG(2)* the dynamic negation of Φ is sequenced, then the interpretation must be that Φ does not preserve truth under a *weakening* of information expressed by the disjunction with the *negation* of the contents of further discourse. More formal, the interpretation of the sequence $\sim\Phi\,;\Psi$ must be that Φ does *not* have the property $\lambda p\ ({}^{\vee}p \vee \neg\Psi(R))$ (where R, again, figures as the landing site for further discourse continuation).

For the sake of compositionality, dynamic negation must be defined in such a way that if an extension with the contents of further discourse is in the dynamic scope of the negation, then it takes the form of a weakening of information with the negation of the contents of further discourse. The following definition of the dual captures this transformation:

Definition 2.9 (Dual) $R_{\langle s,\tau\rangle}{}^{*} = {}^{\wedge}\lambda p\ \neg{}^{\vee}R({}^{\wedge}\ \neg{}^{\vee}p)$, p not free in R

Fact 2.5 shows that the dual turns an extension of information into the required weakening:

Fact 2.5 $({}^{\wedge}\lambda p\ ({}^{\vee}p \wedge \psi))^{*} = {}^{\wedge}\lambda p\ \neg(\neg{}^{\vee}p \wedge \psi) = {}^{\wedge}\lambda p\ ({}^{\vee}p \vee \neg\psi)$

Now we can turn to the definition of dynamic negation itself. The dynamic negation of a formula Φ denotes the properties of propositions whose duals are not in Φ:

Definition 2.10 (Dynamic negation(2)) $\sim\Phi = \lambda R\ \neg\Phi(R^{*})$, R not free in Φ

So the dynamic negation of a dynamic formula brings about a main negation over the formula, and two more negations that enclose the variable R over which abstraction takes place. These two internal negations, so to speak, constitute a context of reversed monotonicity within the downward entailing context induced by the main negation, and this local context of reversed monotonicity is reserved precisely for material that stands outside of the syntactic scope of the negation, but that lands inside its scope by λ-conversion. The following facts are easily proved:

Fact 2.6

1. $R^{**} = R$
2. $\sim\sim\Phi = \Phi$
3. $\downarrow\sim\Phi = \neg\downarrow\Phi$
4. $\sim\uparrow\phi = \uparrow\neg\phi$

The facts 2.6.2 and 2.6.4 show that the definition of dynamic negation in 2.10 satisfies the requirements we set out at the end of section 1.3. An immediate consequence is that dynamic negation as such is not associative with dynamic conjunction.

The definitions of implication, disjunction and universal quantification remain the same, but now they depend on the *DMG(2)* interpretation of ; and $\mathcal{E}d$ and on the definition of dynamic negation. The following equivalences remain valid in *DMG(2)*:

Fact 2.7 (Associativity(2))

1. $[\Phi\,;\Psi]\,;\Upsilon = \Phi\,;[\Psi\,;\Upsilon]$
2. $[\mathcal{E}d\Phi]\,;\Psi = \mathcal{E}d[\Phi\,;\Psi]$
3. $[\Phi \Rightarrow \Psi]\,;\Upsilon = \Phi \Rightarrow [\Psi\,;\Upsilon]$
4. $[\Phi\ \textbf{or}\ \Psi]\,;\Upsilon = \Phi\ \textbf{or}\ [\Psi\,;\Upsilon]$
5. $[\mathcal{A}d\Phi]\,;\Psi = \mathcal{A}d[\Phi\,;\Psi]$

Fact 2.7 indicates that the dynamic potential of *DMG(2)* equals that of *DMG*. Dynamic negation also takes scope over further discourse, as in the extended version of *DMG*, but not over the continuation of the discourse itself, but over its dual now:

Fact 2.8 (Dual associativity) $\sim\Phi\,;\Psi = \sim[\Phi \textbf{ or } \sim\Psi]$

Notice that fact 2.8 is reminiscent of the classical equivalence $\neg\phi \wedge \psi = \neg(\phi \vee \neg\psi)$. Fact 2.9 next shows that dynamic negation behaves like its classical counterpart:

Fact 2.9

1. $\sim\sim\Phi = \Phi$
2. $\sim\mathcal{E}d\Phi = \mathcal{A}d\sim\Phi$
3. $\sim\mathcal{A}d\Phi = \mathcal{E}d\sim\Phi$
4. $\sim[\Phi\,;\Psi] = \Phi \Rightarrow \sim\Psi$
5. $\sim[\Phi \Rightarrow \Psi] = \Phi\,;\sim\Psi$
6. $\sim[\Phi \textbf{ or } \Psi] = \sim\Phi\,;\sim\Psi$

Finally, the closure properties remain the same. We repeat them here:

Fact 2.10

1. $\downarrow\uparrow\phi = \phi$
2. $(\uparrow\phi)(\psi) = \uparrow(\phi(\downarrow\psi))$
3. $\downarrow\sim\Phi = \neg\downarrow\Phi$
4. $\downarrow\mathcal{E}d\Phi = \exists d\downarrow\Phi$
5. $\downarrow\mathcal{A}d\Phi = \forall d\downarrow\Phi$
6. $\downarrow[\uparrow\phi\,;\Psi] = \phi \wedge \downarrow\Psi$
7. $\downarrow[\uparrow\phi \Rightarrow \Psi] = \phi \rightarrow \downarrow\Psi$
8. $\downarrow[\uparrow\phi \textbf{ or } \Psi] = \phi \vee \downarrow\Psi$

Summarizing the facts 2.6–2.10, we conclude that the dynamic negation defined in 2.10 has the dynamic properties that we required it to have above and that, apart from that, it has the properties of classical negation. We now show some applications of *DMG(2)* with dynamic negation. We consider the *DMG(2)*-treatment of (simple variants of) the examples that posed a problem to the extended version of *DMG*. From now on, we let *DMG(2)* refer to the extended version, and the original version presented in section 2.1 we will denote as the restricted version of *DMG(2)*. We start with Groenendijk and Stokhof's example 10.

NO PLAYER LEAVES. HE STAYS
Application and reduction yields the following translation of 10:

$$\sim\mathcal{E}d_i[\uparrow\mathtt{player}(d_i)\,;\uparrow\mathtt{leave}(d_i)]\,;\uparrow\mathtt{stay}(d_i)$$

The truth-conditions of this formula are determined as follows:

$$\begin{gathered}
\downarrow[\sim\mathcal{E}d_i[\uparrow\mathtt{player}(d_i)\,;\uparrow\mathtt{leave}(d_i)]\,;\uparrow\mathtt{stay}(d_i)] \Longleftrightarrow \\
\downarrow[\mathcal{A}d_i\sim[\uparrow\mathtt{player}(d_i)\,;\uparrow\mathtt{leave}(d_i)]\,;\uparrow\mathtt{stay}(d_i)] \Longleftrightarrow \\
\downarrow[\mathcal{A}d_i[\uparrow\mathtt{player}(d_i) \Rightarrow \sim\uparrow\mathtt{leave}(d_i)]\,;\uparrow\mathtt{stay}(d_i)] \Longleftrightarrow \\
\downarrow[\mathcal{A}d_i[\uparrow\mathtt{player}(d_i) \Rightarrow \uparrow\neg\mathtt{leave}(d_i)]\,;\uparrow\mathtt{stay}(d_i)] \Longleftrightarrow \\
\downarrow\mathcal{A}d_i[\uparrow\mathtt{player}(d_i) \Rightarrow [\uparrow\neg\mathtt{leave}(d_i)\,;\uparrow\mathtt{stay}(d_i)]] \Longleftrightarrow \\
\forall d_i(\mathtt{player}(d_i) \rightarrow (\neg\mathtt{leave}(d_i) \wedge \mathtt{stay}(d_i)))
\end{gathered}$$

(By means of the facts 2.9 (twice), 2.6, 2.7 and 2.10.)

NO FARMER BEATS A DONKEY HE OWNS. HE DOESN'T KICK IT EITHER
Application and reduction yields the following translation of 12 (the negation is interpreted as *IV* negation):

$$\sim\mathcal{E}d_i[\uparrow\mathtt{farmer}(d_i)\,;\mathcal{E}d_j[\uparrow\mathtt{donkey\_of}(d_i)(d_j)\,;\uparrow\mathtt{beat}(d_j)(d_i)]]\,;\sim\uparrow\mathtt{kick}(d_j)(d_i)$$

The truth-conditions of this formula are the following:

$$\begin{gathered}\downarrow[\sim\mathcal{E}d_i[\uparrow\mathtt{farmer}(d_i)\,;\mathcal{E}d_j[\uparrow\mathtt{donkey\_of}(d_i)(d_j)\,;\uparrow\mathtt{beat}(d_j)(d_i)]]\,;\sim\uparrow\mathtt{kick}(d_j)(d_i)] \Longleftrightarrow\\ \downarrow[\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i)\Rightarrow\sim\mathcal{E}d_j[\uparrow\mathtt{donkey\_of}(d_i)(d_j)\,;\uparrow\mathtt{beat}(d_j)(d_i)]]\,;\sim\uparrow\mathtt{kick}(d_j)(d_i)] \Longleftrightarrow\\ \downarrow[\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i)\Rightarrow\mathcal{A}d_j[\uparrow\mathtt{donkey\_of}(d_i)(d_j)\Rightarrow\sim\uparrow\mathtt{beat}(d_j)(d_i)]]\,;\sim\uparrow\mathtt{kick}(d_j)(d_i)] \Longleftrightarrow\\ \downarrow[\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i)\Rightarrow\mathcal{A}d_j[\uparrow\mathtt{donkey\_of}(d_i)(d_j)\Rightarrow\uparrow\neg\mathtt{beat}(d_j)(d_i)]\,;\uparrow\neg\mathtt{kick}(d_j)(d_i)] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{farmer}(d_i)\Rightarrow\mathcal{A}d_j[\uparrow\mathtt{donkey\_of}(d_i)(d_j)\Rightarrow[\uparrow\neg\mathtt{beat}(d_j)(d_i)\,;\uparrow\neg\mathtt{kick}(d_j)(d_i)]]] \Longleftrightarrow\\ \forall d_i(\mathtt{farmer}(d_i)\rightarrow\forall d_j(\mathtt{donkey\_of}(d_i)(d_j)\rightarrow(\neg\mathtt{beat}(d_j)(d_i)\wedge\neg\mathtt{kick}(d_j)(d_i))))\end{gathered}$$

(By means of the facts 2.9 (twice), 2.6, 2.7 and 2.10.)

EITHER THERE IS A BATHROOM DOWNSTAIRS, OR IT IS UPSTAIRS.
This sentence is equivalent to the sentence *If it is not the case that there is a bathroom downstairs, then it is upstairs.* In normal form, the translation of both sentences reads as follows:

$$\sim\mathcal{E}d_i[\uparrow\mathtt{bathroom}(d_i)\,;\uparrow\mathtt{downstairs}(d_i)]\Rightarrow\uparrow\mathtt{upstairs}(d_i)$$

The truth-conditions of this formula are the following:

$$\begin{gathered}\downarrow[\sim\mathcal{E}d_i[\uparrow\mathtt{bathroom}(d_i)\,;\uparrow\mathtt{downstairs}(d_i)]\Rightarrow\uparrow\mathtt{upstairs}(d_i)] \Longleftrightarrow\\ \downarrow\sim[\sim\mathcal{E}d_i[\uparrow\mathtt{bathroom}(d_i)\,;\uparrow\mathtt{downstairs}(d_i)]\,;\sim\uparrow\mathtt{upstairs}(d_i)] \Longleftrightarrow\\ \downarrow\sim\sim[\mathcal{E}d_i[\uparrow\mathtt{bathroom}(d_i)\,;\uparrow\mathtt{downstairs}(d_i)]\ \mathbf{or}\ \sim\sim\uparrow\mathtt{upstairs}(d_i)] \Longleftrightarrow\\ \downarrow[\mathcal{E}d_i[\uparrow\mathtt{bathroom}(d_i)\,;\uparrow\mathtt{downstairs}(d_i)]\ \mathbf{or}\ \uparrow\mathtt{upstairs}(d_i)] \Longleftrightarrow\\ \downarrow\mathcal{E}d_i[\uparrow\mathtt{bathroom}(d_i)\,;[\uparrow\mathtt{downstairs}(d_i)\ \mathbf{or}\ \uparrow\mathtt{upstairs}(d_i)]] \Longleftrightarrow\\ \exists d_i(\mathtt{bathroom}(d_i)\wedge(\mathtt{downstairs}(d_i)\vee\mathtt{upstairs}(d_i)))\end{gathered}$$

(By means of definition 2.5, and facts 2.8, 2.6, 2.7 and 2.10.)

IF A CHESSBOX DOESN'T CONTAIN A SPARE PAWN, IT IS TAPED ON TOP OF IT.
The last example that we discuss has the following translation:

$$\mathcal{E}d_i[\uparrow\mathtt{c\_box}(d_i)\,;\sim\mathcal{E}d_j[\uparrow\mathtt{s\_pawn}(d_j)\,;\uparrow\mathtt{in}(d_i)(d_j)]]\Rightarrow\uparrow\mathtt{on}(d_i)(d_j)$$

Truth-conditions are determined as follows:

$$\begin{gathered}\downarrow[\mathcal{E}d_i[\uparrow\mathtt{c\_box}(d_i)\,;\sim\mathcal{E}d_j[\uparrow\mathtt{s\_pawn}(d_j)\,;\uparrow\mathtt{in}(d_i)(d_j)]]\Rightarrow\uparrow\mathtt{on}(d_i)(d_j)] \Longleftrightarrow\\ \downarrow[\mathcal{E}d_i[\uparrow\mathtt{c\_box}(d_i)\,;\mathcal{A}d_j[\uparrow\mathtt{s\_pawn}(d_j)\Rightarrow\sim\uparrow\mathtt{in}(d_i)(d_j)]]\Rightarrow\uparrow\mathtt{on}(d_i)(d_j)] \Longleftrightarrow\\ \downarrow\sim[\mathcal{E}d_i[\uparrow\mathtt{c\_box}(d_i)\,;\mathcal{A}d_j[\uparrow\mathtt{s\_pawn}(d_j)\Rightarrow\sim\uparrow\mathtt{in}(d_i)(d_j)]]\,;\sim\uparrow\mathtt{on}(d_i)(d_j)] \Longleftrightarrow\\ \downarrow\sim\mathcal{E}d_i[\uparrow\mathtt{c\_box}(d_i)\,;\mathcal{A}d_j[\uparrow\mathtt{s\_pawn}(d_j)\Rightarrow[\sim\uparrow\mathtt{in}(d_i)(d_j)\,;\sim\uparrow\mathtt{on}(d_i)(d_j)]]] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{c\_box}(d_i)\Rightarrow\sim\mathcal{A}d_j[\uparrow\mathtt{s\_pawn}(d_j)\Rightarrow[\sim\uparrow\mathtt{in}(d_i)(d_j)\,;\sim\uparrow\mathtt{on}(d_i)(d_j)]]] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{c\_box}(d_i)\Rightarrow\mathcal{E}d_j[\uparrow\mathtt{s\_pawn}(d_j)\,;\sim[\sim\uparrow\mathtt{in}(d_i)(d_j)\,;\sim\uparrow\mathtt{on}(d_i)(d_j)]]] \Longleftrightarrow\\ \downarrow\mathcal{A}d_i[\uparrow\mathtt{c\_box}(d_i)\Rightarrow\mathcal{E}d_j[\uparrow\mathtt{s\_pawn}(d_j)\,;[\uparrow\mathtt{in}(d_i)(d_j)\ \mathbf{or}\ \uparrow\mathtt{on}(d_i)(d_j)]]] \Longleftrightarrow\\ \forall d_i(\mathtt{c\_box}(d_i)\rightarrow\exists d_j(\mathtt{s\_pawn}(d_j)\wedge(\mathtt{in}(d_i)(d_j)\vee\mathtt{on}(d_i)(d_j))))\end{gathered}$$

(By means of the fact 2.9, definition 2.5, fact 2.7, fact 2.9 again (twice), and definition 2.5 and fact 2.10.)

2.3 Discussion

It may be useful to take a break here, and recapitulate our findings thusfar. We have sketched Groenendijk and Stokhof's *DMG* and we proposed an alternative equivalent to it: the restricted version of *DMG(2)*. Furthermore, we considered Groenendijk and Stokhof's extended dynamic version of *DMG*, and we indicated that an equivalent alternative in the format of *DMG(2)* is equally definable. Finally, we proposed our own strongly dynamic version of *DMG(2)*, that is based on an adequate notion of dynamic negation. In fact, then, three different systems are at stake here. In order to conclude this part of the paper, we briefly indicate the characteristic properties of these three systems.

The crucial distinction between the three systems on the one hand, and classical, static, theories of interpretation on the other, lies in the associativity-facts, and the non-commutativity of dynamic conjunction, facts that we repeat here:

Fact 2.11 (Associativity, non-commutativity)

1. $[\Phi\,;\Psi]\,;\Upsilon = \Phi\,;[\Psi\,;\Upsilon]$
2. $[\mathcal{E}d\Phi]\,;\Psi = \mathcal{E}d[\Phi\,;\Psi]$
3. $[\Phi\,;\Psi] \neq [\Psi\,;\Phi]$

(Fact 2.11 also distinguishes all three dynamic Montague grammars from standard *DRT*, because of the compositional treatment of the dynamics.) The three systems also have the following closure properties in common:

Fact 2.12

1. $\downarrow\uparrow\phi = \phi$
2. $\downarrow\lambda T_{\uparrow a}\Phi = \lambda T_a\downarrow[\uparrow T/T]\Phi$; $(\uparrow\phi)(\Psi) = \uparrow(\phi(\downarrow\Psi))$
3. $\downarrow{}^\wedge\Phi = {}^\wedge\downarrow\Phi$; ${}^\vee\uparrow\phi = \uparrow{}^\vee\phi$
4. $\downarrow{\sim}\Phi = \neg\downarrow\Phi$; $\downarrow\mathcal{E}d\Phi = \exists d\downarrow\Phi$
5. $\downarrow[\uparrow\phi\,;\Psi] = \phi \wedge \downarrow\Psi$

The differences between the three systems lie in the ways in which closed dynamic expressions further reduce to *DIL*-formulas without dynamic operators. A complete reduction from *DMG*-formulas in normal form to proper *DIL*-formulas, we can obtain by means of fact 2.11, fact 2.12 and the definition of *DMG*-negation, which is repeated here (an expression is in normal form if $\mathcal{A}d$, $\Rightarrow$ and **or** are replaced by $\mathcal{E}d$, $\sim$ and ;, and if all possible ${}^{\vee\wedge}$-eliminations and λ-conversions have been executed, the last possibly after renaming of variables):

Fact 2.13 ${\sim}\Phi = \uparrow\neg\downarrow\Phi$

Fact 2.14 If Φ is a *DMG*-expression in normal form, without free variables, then $\downarrow\Phi$can be reduced to a proper *DIL*-expression by means of 2.11, 2.12 and 2.13.

. In the appendix we prove this fact. Fact 2.14 indicates that the associativity and non-commutativity expressed by 2.11 in fact is the *only* difference between *DMG* and static theories.

In the extended version of *DMG*, a full, meaning-preserving reduction is possible if instead of 2.13, the associativity of negation is used:

Fact 2.15 ${\sim}\Phi\,;\Psi = {\sim}[\Phi\,;\Psi]$

Fact 2.16 If Φ is an extended *DMG*-expression in normal form, without free variables, then $\downarrow\Phi$can be reduced to a proper *DIL*-expression by means of 2.11, 2.12 and 2.15.

The reduction fact 2.16 implies that the facts 2.11 and 2.15 constitute the only difference between the extended version of *DMG* and traditional theories. The associativity of $\sim$ with ;, we argued, is awfull on the one hand, but it generates the law of double negation on the other hand, and it underlies the associativity of $\mathcal{A}d$, $\Rightarrow$ and **or** with ;, all of which is pleasant. On the remaining first hand, 2.15 makes a mess again in case of triple negation, or in case a negation is added to or deleted from a dynamic universal quantification, a dynamic implication, or a dynamic disjunction, that was treated successfully in extended *DMG*.

A complete reduction in *DMG(2)*, finally, draws from the following facts:

Fact 2.17

1. $\sim[\Phi\,;\Psi]\,;\Upsilon = \sim\Phi\,;\sim[\sim\Psi\,;\Upsilon]$
2. $\sim[\mathcal{E}d\Phi]\,;\Psi = \sim\mathcal{E}d\sim[\sim\Phi\,;\Psi]$
3. $\sim\sim\Phi = \Phi$
4. $\sim\uparrow\phi = \uparrow\neg\phi$

Fact 2.18 If Φ is a *DMG(2)*-expression in normal form, without free variables, then $\downarrow\Phi$can be reduced to a proper *DIL*-expression by means of 2.11, 2.12 and 2.17.

The first two facts in 2.17 guarantee associativity of $\mathcal{A}d$, $\Rightarrow$ and **or** with ;. The two facts clearly exhibit the dual interpretation of further discourse processing within the dynamic scope of the negation. That the last two facts in 2.17 hold, we argued, is a basic requirement for a proper notion of dynamic negation. The last fact, notably, is an ensurance against the plain associativity of $\sim$ with dynamic conjunction that troubled the extended version of *DMG*.

Now that we have come to this point, there seem to be two ways to go. The first one is to stick to *DMG*'s compositionalization of *DRT*. True, the restricted version of *DMG(2)* is equivalent with *DMG* itself, but if one doesn't agree with the extended dynamic account of examples 4–15, one better translates sentences in the most perspicuous way into *DIL*-expressions of the lowest possible type, that is, then it is better to use *DMG* instead of *DMG(2)*. And, of course, there *is* reason not to agree with the extended dynamic account, because the examples under consideration are quite special in a sense, and because the extended dynamic approach is not without problems after all, as we will see shortly.

The other way to go is to embrace *DIL*'s potential to account for a dynamic interpretation of *all* logical operators. We hope to have shown that, if this option is taken, then it is better to use *DMG(2)*, and to translate sentences as generalized quantifiers over propositions, instead of as sets of propositions.

However, if this last line is chosen, two problems have to be solved that pertain to the extended dynamic theories. We end this section by briefly touching on these two problems.

Among the surprising equivalences in *DMG(2)* (and in the extended version of *DMG*), we find two facts that we did not explicitly drew attention to up to now. They are the logical mirror images of equivalences that we used frequently above, and that we repeat here in the clauses 1 and 2 of fact 2.19:

Fact 2.19

1. $\mathcal{E}d\Phi \Rightarrow \Psi = \mathcal{A}d[\Phi \Rightarrow \Psi]$
2. $[[\Phi\,;\Psi] \Rightarrow \Upsilon] = [\Phi \Rightarrow [\Psi \Rightarrow \Upsilon]]$
3. $\mathcal{A}d\Phi \Rightarrow \Psi = \mathcal{E}d[\Phi \Rightarrow \Psi]$

4. $[[\Phi \Rightarrow \Psi] \Rightarrow \Upsilon] = [\Phi ;[\Psi \Rightarrow \Upsilon]]$

Fact 2.19.1 is a typical donkey fact, and fact 2.19.2 is classical. However, their dynamic reverses 2.19.3 and 2.19.4 are not classical, and there seems to be no linguistic support for them. Still, we used facts supporting these equivalences in the interpretation of the examples 13, 14 and 15 above. These facts are puzzling indeed. Whereas our example 14, (repeated here as 16) seems perfectly acceptable, its alleged equivalent 17 seems not:

(16) If there is no bathroom downstairs, then it is upstairs.

(17) If every bathroom is not downstairs, then it is upstairs.

We will come back to this problem in section 4.

Another problem with extended *DMG* and *DMG(2)* is its dynamic rigidity. By this we mean the following. As they are defined presently, all logical operators are assigned a dynamic interpretation and this, of course, is not what we want in general. Quite the reverse, one might say: negation, universal quantification, implication and disjunction in general should be given their static, *DMG*, interpretation, and only in a very limited number of cases a dynamic interpretation of them seems appropriate. Now, although we can derive a static interpretation of these operators from their dynamic interpretation by means of the closure operation $\uparrow\downarrow$, still it seems somewhat counterintuitive that the most frequently used interpretations, the static ones, derive from the rarely used dynamic ones.

In the next section we will sort of tackle this last problem. We will argue that dynamic interpretations of logical operators *can* be derived from static interpretations, this in a flexible dynamic Montague grammar. The transition from a rigid to a flexible semantics is inspired by the observation that the primary instance of *DMG(2)*'s raising operation corresponds to a theorem in flexible calculi such as the Lambek calculus **L**, the Lambek calculus with permutation **LP**, and all kinds of calculi in between, beyond, and next to these two. For that reason, we pick out, in the next section, the most elaborate calculus that suits our purposes, which is Hendriks' system of type change [15], [16]. Hendriks' calculus is a semantically oriented system that can be seen as a subsystem in between **L** and **LP**. In [16], this system is incorporated in a Montague grammar.

We will show that a slight variant of Hendriks' system of type change allows it to extend to discourse phenomena, and that a structural change in the interpretation of type changes of downward monotonic expressions makes it cover all examples discussed in this paper, and even more. We must, however, issue a warning here. The proposals made in the next section are not completely definitive. The type changing calculus is in some respects a too powerfull generator of readings for sentences, and future research is needed to keep these under control. Furthermore, there are some typical side-effects of our use of the type changing system, which we will address in the final section of this paper.

3 Type change and dynamic interpretation

From now on we take a somewhat shifted perspective on the dynamics of discourse. The dynamics will no longer be built into the system of interpretation, but it will appear as a derived property, originating from the possibility to assign scope bearing elements in discourse a number of different scopes. Here, we build upon Hendriks' flexible treatment of quantifier scope phenomena. Hendriks' system is devised to account for scope ambiguities at the sentential level and we will transpose it to the intersentential level of discourse.

One thing we have to point out before we start being flexible. We will use Hendriks' system of type change within the framework of *DIL*, and not in that of

IL, as Hendriks does. The reason is obvious. When in Hendriks' system a quantifier is given wide scope over a certain part of a text, it will not bind pronouns occurring freely there. With *DIL*, however, a quantifier's scope coincides with the domain where it has binding potential. (Some confusion might arise about what we mean by scope here. Strictly speaking, the scope of a quantifier is its syntactic scope. On a more liberal reading, the scope of a quantifier consists of the parts of text that land in its syntactic scope by λ-conversion, possibly after renaming of variables. This we call a quantifier's dynamic scope. Clearly, then, a quantifier in *DIL* can bind pronouns in its dynamic scope, whereas in *IL* it doesn't.)

In this section, we show that by using both type flexibility and *DIL*, we can capture all *DMG(2)*-results in a compositional way, this on the basis of the simplest basic translations of atomic expressions. We proceed as follows. We first present a slight modification of Hendriks' system of type change, and indicate how it can be used to account for scope ambiguities. Next, we extend the language with categorematic sentential connectives with a most intuitive basic translation. This already paves the way for dynamic interpretation. A real hurdle has to be taken in section 3.3, where we readdress the issue of downward monotonicity and wide scope. We propose an amended interpretation of the type changing operations that act upon downward monotonic expressions, and this amendation largely consists of a generalization of the dual operation that we presented in section 2.2. Finally, we show that the resulting flexible Montague grammar extends the descriptive power of *DMG(2)*. Some more puzzling examples involving wide scope of downward monotonic operators are treated to our satisfaction.

3.1 A flexible account of scope phenomena

In a flexible Montague grammar, syntactic categories are assigned the most simple basic types. For instance, the basic type assigned to the category of *NP*'s is the type e of individuals. Now, since intransitive verb phrases denote functions from individual concepts to truth values, these need not be of a basic category. Therefore, the basic categories in a flexible Montague grammar are the categories S, CN and NP. Derived categories are of the form A/B and $B\backslash A$. Abbreviations used in the fragment are IV ($= NP\backslash S$), Det ($= NP/CN$), and TV ($= IV/NP$). We redefine the function f from categories to basic *DIL*-types:

Definition 3.1 (Basic category to type assignment)

1. $f(S) = t$; $f(NP) = e$; $f(CN) = \langle\langle s,e\rangle,t\rangle$
2. $f(A/B) = f(B\backslash A) = \langle\langle s, f(B)\rangle, f(A)\rangle$

As appears from definition 3.1, the basic types assigned to syntactic categories are essentially simpler than they are in traditional Montague grammar. Like we said, the type of *NP*'s is as simple as can be, and the type of *TV*'s is now basically the type of relations between individual concepts instead of between individual concepts and quantifiers over individual concepts. Of course, the use of basic types will not suffice for the interpretation of a comprehensive fragment of natural language. Notably, a treatment of quantifying noun phrases does not fit in, and requires an extension of the fragment. In [16], Hendriks presents a system of type change, with associated interpretation, that allows one to deal with quantifying noun phrases, and with their scope, on the basis of the category to type assignment in 3.1. The type changing system is presented as an alternative to and an improvement of other approaches to quantifier scope phenomena. Here, we will not discuss the relative merits of different approaches to quantifier scope phenomena, but we present Hendriks' system right away.

There are at least two phenomena that a treatment of quantification in natural language must account for. First, it must explain the conjoinability of proper names, whose type is e, intuitively, and quantifying noun phrases, whose type in a compositional theory of interpretation must minimally be $\langle\langle e,t\rangle,t\rangle$. Second, it must account for scope ambiguity such as exhibited in the sentence *Every man loves a woman.* This sentence can be taken to mean that for every man there is a (possibly different) woman that he loves, and it can mean that there is a woman whom every man is in love with. Both phenomena are dealt with in Hendriks' flexible Montague grammar by letting the interpretations of expressions shift through types. Before we state the required type shifts, it is useful to define some abbreviations:

Notation convention 1

1. If $\vec{a}^n$ is a sequence of types $a_1,\ldots,a_n$, then $\langle\vec{a}^n,b\rangle = \langle a_1,\ldots,\langle a_n,b\rangle\ldots\rangle$
2. If $\vec{x}^n$ is a sequence of variables $x_1,\ldots,x_n$, then $\lambda\vec{x}^n\phi = \lambda x_1\ldots\lambda x_n\phi$
3. If $\vec{x}^n$ is a sequence of variables $x_1,\ldots,x_n$, ϕ an expression of type $\langle\vec{a}^n,b\rangle$ where x_i is of type a_i $(1{\leq}i{\leq}n)$, then $\phi(\vec{x}^n) = \phi(x_1)\ldots(x_n)$

When we use the notation of generalized abstraction and application, we include abstraction over and application of states, as if these constitute a type of their own. Clearly, the loosely formulated expressions $\lambda x_s\phi$ and $\phi(x_s)$ must be read as ${}^\wedge\phi$ and ${}^\vee\phi$ then. Furthermore, in the sequel we also use the abbreviation ${}^\frown a$ for the type $\langle s,a\rangle$.

In order to capture the idea that expressions are subject to type shifts, each syntactic category C is not associated with one type, but with a set of types $T(C)$, the type set of C. This set contains the basic type assigned to C, and types derived from the basic type by the following type changing rules:

Definition 3.2 (Type set) The type set of C, $T(C)$ is the smallest set such that:

1. $f(C) \in T(C)$
2. If $\langle\vec{a}^i,b\rangle \in T(C)$, then $\langle\vec{a}^i,\langle{}^\frown\langle{}^\frown b,t\rangle,t\rangle\rangle \in T(C)$
3. If $\langle\vec{a}^i,\langle b,\langle\vec{c}^j,t\rangle\rangle\rangle \in T(C)$, then $\langle\vec{a}^i,\langle{}^\frown\langle{}^\frown\langle b,t\rangle,t\rangle,\langle\vec{c}^j,t\rangle\rangle\rangle \in T(C)$
4. If $\langle\vec{a}^i,\langle{}^\frown\langle\vec{b}^j,\langle\vec{d}^n,t\rangle\rangle,\langle\vec{d}^n,t\rangle\rangle\rangle \in T(C)$, then $\langle\vec{a}^i,\langle{}^\frown\langle\vec{b}^j,\langle c,\langle\vec{d}^n,t\rangle\rangle\rangle,\langle c,\langle\vec{d}^n,t\rangle\rangle\rangle\rangle \in T(C)$

The second clause of this definition is related to the rule of value raising ([VR]) defined below. By means of this rule we can raise an object of type b into a generalized quantifier over objects of type b. A crucial example is the raising of the value of individual denoting expressions like proper names. If the type of a proper name is raised, it has the same type as that of quantifying noun phrases. Another important application is the raising of the value of sentences. As we will see below, raising the value of a sentence corresponds to the dynamic lift of *DMG(2)*. The third clause in definition 3.2 corresponds to argument raising ([AR]). This type shift allows a function from objects of type b to apply to generalized quantifiers over objects of type b. More specific, on the basis of this type shift the basic translation of an extensional transitive verb can be made to apply to two quantifying argument expressions. And what is more, it can be made to apply to the two quantifying arguments in two different ways, and the two ways of making it applicable correspond to two different scopings of the two quantifiers. With the last clause in definition 3.2 we associate the rule of generalized division ([GD]). This rule allows a quantifying expression to inherit possible extra argument slots from the function to which it is made to apply. The application of a divided quantifier to such a function, in fact, will come down to the composition of both functions.

We now turn to some examples of basic translations of expressions in the fragment. A basic translation of an expression of a certain category is of the basic type assigned to that category, or of a type derivable from the basic type by type change. In what follows x and y are variables of type $\langle s, e\rangle$, and P and Q of type $\langle s, \langle\langle s, e\rangle, t\rangle\rangle$; $\mathtt{j}$ is a constant of type e, $\mathtt{man}$ and $\mathtt{walk}$ are constants of type $\langle e, t\rangle$, and $\mathtt{love}$ of type $\langle e, \langle e, t\rangle\rangle$:

Definition 3.3 (Translations of basic expressions)

1. $man \rightsquigarrow \lambda x\ \mathtt{man}({}^{\vee}x)$
2. $walk \rightsquigarrow \lambda x\ \mathtt{walk}({}^{\vee}x)$
3. $love \rightsquigarrow \lambda y \lambda x\ \mathtt{love}({}^{\vee}y)({}^{\vee}x)$
4. $a_i \rightsquigarrow \lambda P \lambda Q\ \exists d_i({}^{\vee}P({}^{\wedge}d_i) \wedge {}^{\vee}Q({}^{\wedge}d_i))$
5. $every_i \rightsquigarrow \lambda P \lambda Q\ \forall d_i({}^{\vee}P({}^{\wedge}d_i) \rightarrow {}^{\vee}Q({}^{\wedge}d_i))$
6. $no_i \rightsquigarrow \lambda P \lambda Q\ \neg\exists d_i({}^{\vee}P({}^{\wedge}d_i) \wedge {}^{\vee}Q({}^{\wedge}d_i))$
7. $he_i \rightsquigarrow d_i$
8. $John_i \rightsquigarrow \mathtt{john}$

(As appears from the last clause, we simply treat a proper name as an individual denoting expression. This does not enable proper names to figure as the antecedent of a pronoun. Therefore, we assume that at the beginning of a discourse the value of certain discourse markers is set to the interpretations of proper names. But of course, an alternative might be to translate a proper name like *John* itself into the existentially quantifying noun phrase *an individual who is John*, as in definition 1.11.)

So we see that natural language expressions are assigned simple basic interpretations. However, the meanings of natural language expressions in flexible Montague grammar are not exhausted by their basic translations. Natural language expressions take interpretations from various differently typed domains, in accordance with the type shifts that operate on them. Therefore, the meaning of an expression will be a set of interpretations which have types derivable from the basic type assigned to the expression's category. The interpretation set of an expression is determined by its basic translation *and* derivable translations, where the notion of derivable translation is defined as follows (in this definition $n \geq 1$):

Definition 3.4 (Derivable translations)

[Identity] $\phi \Longrightarrow \phi$

[nVR] If ϕ is an expression of type $\langle \vec{a}^{n-1}, b\rangle$, Y a variable of type ${}^{\frown}\langle {}^{\frown}b, t\rangle$, and $\vec{x}^{n-1}$ and Y are not free in ϕ, then

$$\phi \Longrightarrow \lambda \vec{x}^{n-1} \lambda Y\ {}^{\vee}Y({}^{\wedge}\ \phi(\vec{x}^{n-1}))$$

[nAR] If ϕ is an expression of type $\langle \vec{a}^{n-1}, \langle b, \langle \vec{c}^{m}, t\rangle\rangle\rangle$, y and Y variables of type b and ${}^{\frown}\langle {}^{\frown}\langle b, t\rangle, t\rangle$ respectively, and $\vec{x}^{n-1}$, Y, $\vec{z}^{m}$ and y are not free in ϕ, then

$$\phi \Longrightarrow \lambda \vec{x}^{n-1} \lambda Y \lambda \vec{z}^{m}\ \ {}^{\vee}Y({}^{\wedge}\lambda y\ \phi(\vec{x}^{n-1})(y)(\vec{z}^{m}))$$

[nGD(c)] If ϕ is an expression of type $\langle \vec{a}^{n-1}, \langle {}^{\frown}\langle \vec{b}^{m}, \langle \vec{d}^{j}, t\rangle\rangle, \langle \vec{d}^{j}, t\rangle\rangle\rangle$, z and Y variables of type c and ${}^{\frown}\langle \vec{b}^{m}, \langle c, \langle \vec{d}^{j}, t\rangle\rangle\rangle$ respectively, and $\vec{x}^{n-1}$, Y and z are not free in ϕ, then

$$\phi \Longrightarrow \lambda \vec{x}^{n-1} \lambda Y \lambda z\ \phi(\vec{x}^{n-1})({}^{\wedge}\lambda \vec{y}^{m}\ {}^{\vee}Y(\vec{y}^{m})(z))$$

[Transitivity] $\phi \Longrightarrow \psi$ if $\phi \Longrightarrow \chi$ and $\chi \Longrightarrow \psi$

(In our exposition of the type changing system we omitted the rule of argument lowering, which is not relevant for the present discussion. On the other hand, we added the rule [GD], labeled generalized division. This rule effectively overlaps with Hendriks' rule of argument raising. In fact, [GD] will eventually replace [AR] for reasons that will come out shortly. For the time being, however, it is convenient to have both rules at our disposal.)

In the definition of derivable translations those labeled [nVR], [nAR] and [nGD(c)] are the crucial ones. These clauses provide the type shifts in the clauses 2–4 of definition 3.2 with an interpretation. (The first and last clause in definition 3.4, [Identity] and [Transitivity], merely serve to oil the wheels of the derivation system.) The rule [nVR] is that of n-th value raising, or raising of the n-th value of an expression, as we will also say. This rule raises the n-th value of an expression, which is of some type b into the type of generalized quantifiers over objects of type b. For instance, if we raise the first value of the translation of *John* by means of [1VR], we get an expression denoting the set of properties of the individual concept of John: $\lambda P\ {}^{\vee}P({}^{\wedge}\mathbf{john})$, which is its interpretation in Montague grammar.

The rule [nAR] is that of n-th argument raising. This rule raises the n-th argument of a function into the type of generalized quantifiers over objects of the original argument type. For instance, the raising of the first argument of an extensional transitive verb R, generates its interpretation in Montague grammar: $\lambda T \lambda x\ {}^{\vee}T({}^{\wedge}\lambda y\ R'(y)(x))$.

Finally we have the rule labeled [nGD(c)], to be phrased as 'the division of the n-th argument of a function into type c'. Like we said, this rule generates the intensional composition of the original function and the function to which it applies (after $n-1$ other applications). For instance, we can divide the first argument of a quantifying noun phrase into type $\langle s, e\rangle$, and thus make it applicable to relations between individual concepts, instead of to properties of individual concepts. This rule, together with [VR], therefore provides us with another way to generate an interpretation of an intransitive verb phrase that consists of an extensional transitive verb and a quantifying object argument.

We can now state what the meaning is of an expression in flexible Montague grammar. The translation set of an expression is the closure under $\Longrightarrow$ of the singleton set that contains the basic translation of the expression (T' indicates the basic translation of an expression T):

Definition 3.5 (Translation set) The translation set of an expression T, T'', is the smallest set such that

1. $T' \in T''$
2. If $T' \Longrightarrow \beta$, then $\beta \in T''$

The last thing to be defined before we can start working with flexible Montague grammar is the interpretation of application. Since we assign sets of interpretation to expressions, the interpretation of application needs some adjustment. When a functor expression is adjoined to an expression of the appropriate argument category, then we apply members of the interpretation set of the functor expression to 'fitting' members of the interpretation set of the argument expression. Since we have turned to a bidirectional grammar we have two rules of application, right application ([RA]) and left application ([LA]):

Definition 3.6 (Application)

[RA] $T : B/A,\ U : A \Longrightarrow TU : B$ where TU'' is the smallest set such that if $\alpha \in U''$, $\beta \in T''$, and α is of type a, and β of type $\langle {}^{\frown}a, b\rangle$, then $\beta({}^{\wedge}\alpha) \in TU''$

[LA] $U : A,\ T : A\backslash B \Longrightarrow UT : B$ where TU'' is the smallest set such that if $\alpha \in U''$, $\beta \in T''$, and α is of type a, and β of type $\langle \hat{} a, b\rangle$, then $\beta(^\wedge\alpha) \in TU''$

We now show, very sketchy, two applications of flexible Montague grammar. The first example concerns the two scope readings of a sentence consisting of an transitive verb and two quantifying noun phrases. If U and V are quantifying noun phrases, and R an extensional transitive verb, then the interpretation of the sentence URV contains the following two readings, in the first of which the subject has wide scope, in the second the object:

$$([2\mathrm{AR}]([1\mathrm{AR}](R')))(^\wedge V')(^\wedge U') = U'(^\wedge\lambda x\ V'(^\wedge\lambda y\ R'(y)(x)))$$
$$([1\mathrm{AR}]([2\mathrm{AR}](R')))(^\wedge V')(^\wedge U') = V'(^\wedge\lambda y\ U'(^\wedge\lambda x\ R'(y)(x)))$$

The second example concerns wide scope of a quantifier that occurs in an embedded sentence:

(18) Mary believes that a mathematician wrote *Through the Looking Glass*

This sentence has a de re reading which we can account for by type changes on the verbs *believe* and *wrote*:

$$([2\mathrm{AR}]([3\mathrm{VR}](\mathtt{write})))(^\wedge\mathtt{TtLG})(^\wedge\lambda P\ \exists d(\mathtt{math} \wedge {}^\vee P(^\wedge d))) =$$
$$\lambda R\ \exists d(\mathtt{math}(d) \wedge {}^\vee R(^\wedge\mathtt{write}(\mathtt{TtLG})(d)))\ \ (= \alpha)$$

$$([1\mathrm{AR}](\mathtt{believe}))(^\wedge\alpha)(^\wedge\mathtt{mary}) =$$
$$\exists d(\mathtt{math}(d) \wedge \mathtt{believe}(^\wedge\mathtt{write}(\mathtt{TtLG})(d))(\mathtt{mary}))$$

The system of type change is very powerfull. It allows one to generate all the scope configurations between the quantifiers in a part of text that respect the application structure of the text. Furthermore, it generates an infinity of readings of different types, and this is what makes it hard to study the general properties of the system in depth. However, if the system is used to account for specific scope configurations, it is no more incomprehensible than, for instance, the determination of the Elo-rating of the Grand-masters in chess.

We now give an informal characterization of how the type changes in 3.4 can be used to derive specific scope configurations for a part of text on the basis of its (syntactic) application structure. Think of an application structure as a structure that consists of an atomic expression of a functional category together with n argument expressions, where $n \geq 0$. The argument expressions can be atomic expressions as well, but they may also be application structures themselves. We will say that an atomic expression f is the mother of an expression x if f applies to x and, likewise, we then call x a daughter of f. The use of the terms 'mother' and 'daughter' furthermore underlies the use of the terms 'grandmother' and 'aunt' and the like in a straightforward way.

In the fragment of flexible Montague grammar, the daughters of a mother may have a translation of a type that does not match with the type of the relevant argument place of the mother. Notably, the i-th argument place of the translation of a mother may be of type a, whereas the i-th daughter of the mother translates as a quantifier over objects of type a. In that case, we say that the mother has a quantifying daughter as its i-th argument, or, simply, that the i-th daughter of the mother is a quantifier. In such a case, an application of [AR] to the translation of the mother accomodates it to its quantifying daughter, and the result of the raising is that the daughter takes scope over the mother. Still, a daughter can have more. A quantifying daughter may climb up in the application structure so to speak, and take scope over its grand-(grand-...)mother, and over other far relatives.

We will see presently how this comes about by using type changes in the easiest way. Before that, we notice two things. First, a quantifying descendant can only take scope over a far relative if it takes scope over a common ancestor. Second, a quantifying descendant Q_1 may take scope over an ancestor that she shares with another quantifying relative Q_2, but still not have Q_2 in its scope. (Clearly, if two quantifying relatives take scope over a common ancestor, one must be in the scope of the other.)

A specific scope configuration for a given application structure can be derived in the following way. First, for any quantifying i-th daughter, [iAR] must apply to the mother. The order of applications of [AR] on the mother hereby determines the mutual scope of the quantifying daughters: if [iAR] applies after [jAR] applied, then the i-th daughter takes scope over the j-th daughter. Second, quantifying daughters may be raised higher up in the application structure by applying [nVR] to the mother, where n is the number of daughters of the mother plus one. (Other applications of [VR] serve no purpose.) An application of [nVR] on a mother brings it about that the compound family that consists of the mother with its $n-1$ daughters constitutes a quantifying argument again for the grandmother, and the quantifying daughters that come to range (at least) over the grandmother, are those that land in an argument place of the mother that is raised after [nVR] applied. (We may call these the *promoted* daughters.) Third, if quantifying daughters are promoted because [nVR] applied to the mother before the relevant argument was raised, then raisings of the grandmother further determine the scope of the daughters. So, suppose that a mother is the i-th of the m daughters of a grandmother and that some daughters of the mother are promoted by an application of [nVR] to the mother. In that case an application of [iAR] to the grandmother makes her applicable to the compound consisting of the mother and her descendants. And if [iAR] applies to the grandmother after an application of [jAR], then the promoted descendants of the i-th mother take scope over the j-th daughter of the grandmother (or, over the promoted descendants of the j-th daughter). Furthermore, if [iAR] applies to the grandmother after an application of [mVR], then the promoted daughters of the i-th daughter are promoted again, to the level of the grandgrandmother now, and the type changes operating on the grandgrandmother further determine their scope.

Basically, this is all we need: an application of [iAR] to a mother if her i-th daughter is a quantifier; and an application of [jAR] to a grandmother, if [nVR] applied to her j-th daughter (that itself has $n-1$ arguments). However, there is one more complication. If quantifying daughters (or descendants) of a mother are promoted to a higher level in the application structure, then they need not land all at the very same level. For instance, of two promoted quantifying daughters of a mother, the first daughter may take scope over predecessors or quantifying relatives that themselves take scope over the second daughter. This is made possible by another application of [nVR] to the mother *in between* the two applications of [AR] that relate to the two daughters. So, if [nVR] applies to a mother after [jAR] did, but before an application of [iAR], then predecessors of the mother, or other relatives of hers, may take scope over (promoted descendants of) the j-daughter, and still be in the scope of (promoted descendants of) the i-daughter. But, of course, if [nVR] has applied to a mother k times, then the relevant argument of the grandmother must be raised k times as well. More in general then, if a quantifier is promoted since the relevant application of [AR] occurs after l applications of [nVR] to the j-th daughter of a grandmother, then it is the l-th application of [jAR] to the grandmother that settles its scope. So, if the l-th application of [iAR] to a grandmother occurs after the h-th occurrence of [jAR], then the descendants of the i-th daughter that are promoted after l applications of [VR] take scope over the descendants of the j-th daughter that are *not* promoted after h applications of [VR]. We hope that these remarks suffice to understand the sequel. As for an exercise, the reader may check for himself that an application of [nVR] to a mother serves no purpose if it is not followed by an application of [AR], and that the k-th application of [nVR] to the j-th daughter of a grandmother serves no purpose, if the k-th application of [iAR] to the grandmother immediately follows the $k-1$-th application of [iAR] to her. Those who want to know more about the semantic properties of the type changing system are advised to consult Hendriks [16].

So far, we have used only instances of value- and argument raising in the derivation of scope configurations. So, what do we need division for? There are two reasons for adopting this rule. In the first place, with generalized division we have a stronger system. As we will see below, the division rule is crucial when we want to derive the *DMG(2)*-interpretations of the connectives from classical basic interpretations. In the second place, when we have division at our disposal, we can do away with argument raising. (The motivation for dropping [AR] will be given in section 3.3.) We will now sketch how applications of [AR] in the derivation of a scope configuration can be replaced by instances of [VR] and [GD]. Before we show how, we introduce one more notation convention:

Notation convention 2
$[\mathrm{nGD}_{\vec{c}^m}](\phi) = [\mathrm{nGD}(c_1)]\ldots([\mathrm{nGD}(c_m)](\phi))$
$[\mathrm{nVR}_{\langle\vec{c}^m,t\rangle}](\phi) = [\mathrm{nGD}_{\vec{c}^m}]([\mathrm{nVR}](\phi))$

Now consider the following elimination of [1AR]:

Fact 3.1 ([1AR] elimination)

$$\begin{gathered}([\mathrm{1AR}](\beta_{\langle a,\langle\vec{b}^n,t\rangle\rangle}))({}^\wedge\alpha_{\langle{}^\wedge\langle a,t\rangle,t\rangle}) = \lambda\vec{y}^n\ \alpha({}^\wedge\lambda x\ \beta(x)(\vec{y}^n)) = \\ (\lambda T\lambda\vec{y}^n\ {}^\vee T({}^\wedge\beta)(\vec{y}^n))({}^\wedge\lambda P\lambda\vec{y}^n\ \alpha({}^\wedge\lambda x\ {}^\vee P(x)(\vec{y}^n))) = \\ ([\mathrm{1VR}_{\langle\vec{b}^n,t\rangle}](\beta))({}^\wedge[\mathrm{1GD}_{\vec{b}^n}](\alpha))\end{gathered}$$

We will not prove here that we can eliminate all applications of [AR] on the functions in the application structures we sketched above, but we only indicate how this elimination takes place. We distinguish two cases. In the first case [iAR] applies to a mother function in order to accomodate it to a quantifying daughter. Now if [iAR] applies to a function of type $\langle\vec{a}^i,\langle\vec{b}^j,t\rangle\rangle$, then it will be replaced by an application of $[\mathrm{iVR}_{\langle\vec{b}^j,t\rangle}]$, and $[\mathrm{1GD}_{\vec{b}^j}]$ is applied to the quantifying argument. In the second case, a series of k applications of [iAR] applies to a grandmother, in order to accomodate it to an argument which itself is an application structure. Now suppose that the mother of this compound argument is itself applied to $m-1$ daughters. Then the k applications of [iAR] to the grandmother correspond to k applications of [mVR] to the mother. In that case, if the l-th ($l \leq k$) application of [iAR] to the grandmother operates on a function of type $\langle\vec{a}^i,\langle\vec{b}^j,t\rangle\rangle$, then it will be replaced by an application of $[\mathrm{iVR}_{\langle\vec{b}^j,t\rangle}]$, and the l-th application of [mVR] on the mother is replaced by an application of $[\mathrm{mVR}_{\langle\vec{b}^j,t\rangle}]$.

The following example shows how we derive, without [AR], the two possible scopings of the sentence *URV* consisting of transitive verb R and the two quantifying noun phrases U and V (T abbreviates the type of quantifiers $\langle{}^\wedge\langle{}^\wedge e,t\rangle,t\rangle$):

$$\begin{gathered}([\mathrm{2VR}]([\mathrm{1VR}_{\langle{}^\wedge e,t\rangle}](R')))({}^\wedge[\mathrm{1GD}_{{}^\wedge e}](V'))({}^\wedge U') = U'({}^\wedge\lambda x\ V'({}^\wedge\lambda y\ R'(y)(x))) \\ ([\mathrm{1VR}_{\langle{}^\wedge T,t\rangle}]([\mathrm{2VR}](R')))({}^\wedge[\mathrm{1GD}_{{}^\wedge T}](V'))({}^\wedge U') = V'({}^\wedge\lambda x\ U'({}^\wedge\lambda y\ R'(y)(x)))\end{gathered}$$

(We remark once more that we will keep using [AR] for ease of exposition.)

We leave the exposition of flexible Montague grammar here. In the next section we extend Hendriks' flexible Montague grammar to a (rudimentary) discourse grammar. The system that results there we will call flexible dynamic Montague grammar, or *FDMG* for short.

3.2 Flexible dynamic Montague grammar

In order to extend a flexible sentence grammar to a flexible discourse grammar, all we need is a categorematic treatment of sentential (and other) connectives. We propose the following intuitively motivated translations of such connectives (p and q are variables of type $\langle s,t\rangle$, x of type $\langle s,e\rangle$, and P and Q of type $\langle s,\langle\langle s,e\rangle,t\rangle\rangle$):

Definition 3.7 (Basic translations of connectives)

1. $._{S\backslash(S/S)} \rightsquigarrow \lambda p\lambda q\ ({}^{\vee}p \wedge {}^{\vee}q)$
2. $not_{S/S} \rightsquigarrow \lambda p\ \neg{}^{\vee}p$
3. $if_{(S/S)/S} \rightsquigarrow \lambda p\lambda q\ ({}^{\vee}p \rightarrow {}^{\vee}q)$
4. $or_{S\backslash(S/S)} \rightsquigarrow \lambda p\lambda q\ ({}^{\vee}p \vee {}^{\vee}q)$
5. $who_{(CN\backslash CN)/IV} \rightsquigarrow \lambda P\lambda Q\lambda x\ ({}^{\vee}P(x) \wedge {}^{\vee}Q(x))$

Three crucial *DMG(2)*-operations appear to be derivable now from these basic translations by means of type change (R is a variable of type $\langle s,\tau\rangle$, p of type $\langle s,t\rangle$, and Φ and Ψ are expressions of type $\langle\langle s,\tau\rangle,t\rangle$):

Fact 3.2

1. $\uparrow\phi_t = \lambda R\ {}^{\vee}R({}^{\wedge}\phi) = [\mathrm{1VR}]\phi$
2. $\Phi\,;\Psi = \lambda R\ \Phi({}^{\wedge}\lambda p\ {}^{\vee}p \wedge \Psi(R)) = [\mathrm{1AR}]([\mathrm{2GD}_{\hat{}\tau}](.\,'))({}^{\wedge}\Phi)({}^{\wedge}\Psi)$
3. $\Phi\ \textbf{or}\ \Psi = \lambda R\ \Phi({}^{\wedge}\lambda p\ {}^{\vee}p \vee \Psi(R)) = [\mathrm{1AR}]([\mathrm{2GD}_{\hat{}\tau}](or\,'))({}^{\wedge}\Phi)({}^{\wedge}\Psi)$

(Notice that the two sentential connectives . and *or* are the only fully upward monotonic operators from definition 3.7. Type changes of downward monotonic expressions are the subject of the next section.) Fact 3.2 shows that important *DMG(2)*-operations are already present in *FDMG*. Crucial in the derivation of dynamic conjunction and disjunction is the application of the rule of division. The combined application of [2GD] and [1AR] generates the interpretation of sequencing as composition. If we only had [AR] at our disposal, and not [GD], this result could not be established. (Notice that the application of [AR] is eliminable.) The main difference between a system with [GD] and one with [AR], is that sentential modifiers of type $\langle\hat{}t,t\rangle$ in the first system are also treated as some kind of quantifiers (quantifiers over zero-tuples of objects!), that can climb up so to speak in an application structure. This means that, for instance, a sequence function $\lambda q\ (\phi\wedge{}^{\vee}q)$ may extend its (semantic) scope beyond its syntactic argument.

The *DMG(2)*-notions of dynamic existential quantification and of dynamic universal quantification are not present in *FDMG*, since we have no natural language expressions that denote these quantifiers themselves. But of course, if we were to introduce (theoretical) expressions *there is an individual such that* and *for every individual it holds that*, of the category S/S, with type $\langle\hat{}t,t\rangle$, and basic translations $\lambda p\ \exists d({}^{\vee}p)$ and $\lambda p\ \forall d({}^{\vee}p)$ respectively, then the *DMG(2)* counterparts of these genuinely quantifying expressions are derivable as well:

Fact 3.3

1. $\mathcal{E}d\Phi = \lambda R\ \exists d\Phi(R) = [\mathrm{1GD}_{\hat{}\tau}](\lambda p\ \exists d({}^{\vee}p))({}^{\wedge}\Phi)$
2. $\mathcal{A}d\Phi = \lambda R\ \forall d\Phi(R) = [\mathrm{1GD}_{\hat{}\tau}](\lambda p\ \forall d({}^{\vee}p))({}^{\wedge}\Phi)$

What we cannot derive at present is the *DMG(2)*-translation of the determiners *a* and *every*. We can divide the second argument of these determiners into type $\hat{}\tau$ and the second argument then has exactly the same dynamic and truth-conditional properties as in *DMG(2)*. On the other hand, we cannot give the first argument of the determiners dynamic scope over the second *within* the scope of the *DIL*-quantifier in the translation of the determiner, as in *DMG(2)*. If we raise (over) the first argument of the translation of a determiner, then this argument takes scope over the determiner itself. However, the examples below show that this difference between a fully dynamic translation of a determiner in *FDMG*, and its translation in *DMG(2)* need not lead to differences in the readings assigned to sentences with dynamic determiners. In section 4.2 we come back to this issue.

We now turn to a flexible treatment of examples discussed in the dynamic Montague grammars above. In this section we only attend to examples in which no downward monotonic expressions occur, that is, (variants of) examples 1, 7 and 13. In the next subsection we also discuss the issue of flexible scope and downward monotonicity, and treat the remaining examples. We start with example 1, repeated here as 19:

(19) A man walks. He talks.

In order to derive the dynamic reading of this example, we change the type of two lexical items. We first raise the verb *walk* to the level of dynamic formulas, and next raise its first argument, this in order to make it applicable to the translation of the quantifying noun phrase *A man*. After application and reduction, the translation of the first sentence reads:

$$([1\text{AR}]([2\text{VR}](\textit{walk}\,')))(^\wedge\ \textit{a man}\,') = \lambda R\ \exists d(\mathtt{man}(d) \wedge {}^\vee R(^\wedge\ \mathtt{walk}(d)))$$

The translation of the second sentence is obtained by functional application of the two basic translations:

$$\textit{talk}'(^\wedge\ \textit{he}\,') = \mathtt{talk}(d)$$

All we further need is a dynamic interpretation of the sequencing operator. Its first argument is raised, in order to make it applicable to the dynamic translation of the first sentence:

$$[1\text{AR}](.\,') = \lambda P \lambda q\ {}^\vee P(^\wedge \lambda p\ {}^\vee p \wedge {}^\vee q)$$

If we apply this translation of the sequencing operator to the intensions of the translation of the two sequents, we get the following reading of example 19, which equals the interpretation it has in *DMG(2)*:

$$\exists d(\mathtt{man}(d) \wedge \mathtt{walk}(d) \wedge \mathtt{talk}(d))$$

The next example is 7, repeated here as 20:

(20) Every player chooses a pawn. He puts it on square one.

Again, the verb in the first sentence is raised to the dynamic type. Next, because this transitive verb is flanked by two quantifying noun phrases, both of its arguments have to be raised. The following translation of the first sentence of 20 is the result:

$$\begin{gathered}([1\text{AR}]([2\text{AR}]([3\text{VR}](\textit{choose}\,'))))(^\wedge\ a_j\ \textit{pawn}\,')(^\wedge\ \textit{every}_i\ \textit{player}\,') = \\ \lambda R\ \forall d_i(\mathtt{player}(d_i) \rightarrow \exists d_j(\mathtt{pawn}(d_j) \wedge {}^\vee R(^\wedge\ \mathtt{choose}(d_j)(d_i))))\end{gathered}$$

The translation of the second sentence simply reads: $\mathtt{put\_on}(1)(d_j)(d_i)$. If we use the same interpretation of the sequencing operator now as in the example above, the following interpretation of example 20 results:

$$\forall d_i(\mathtt{player}(d_i) \rightarrow \exists d_j(\mathtt{pawn}(d_j) \wedge \mathtt{choose}(d_j)(d_i) \wedge \mathtt{put\_on}(1)(d_j)(d_i)))$$

The last example without downward monotonic expressions that we discussed above is 13, repeated here as 21:

(21) Either there is a bathroom downstairs or it is upstairs.

First we raise the value of the verb in the first sentence, and then raise its first argument in order to let it apply to the subject *a bathroom*:

$$([1\mathrm{AR}]([2\mathrm{VR}](\textit{downstairs}\,'))(^\wedge\ \textit{a bathroom}\,') =$$
$$\lambda R\ \exists d(\mathtt{bathroom}(d) \wedge {}^\vee R(^\wedge\ \mathtt{down}(d)))$$

The translation of the second disjunct reads, a bit simplified, $\mathbf{up}(d)$. So we only need to raise the first argument of the disjunction operator in order to let it apply to the raised translation of the first disjunct:

$$[1\mathrm{AR}](\mathbf{or}\,') = \lambda P \lambda q\ {}^\vee P(^\wedge \lambda p\ {}^\vee p \vee {}^\vee q)$$

If we apply this translation to the intensions of the translation of the two sentences, we get the following reading of example 21, which equals the interpretation it has in *DMG(2)*:

$$\exists d(\mathtt{bathroom}(d) \wedge (\mathtt{down}(d) \vee \mathtt{up}(d)))$$

We see that a flexible treatment of scope phenomena provides the basic tools for dynamic interpretation, that is, if *DIL* is the framework of interpretation, and if we disregard downward monotonicity for the moment. Formulas can be raised into the dynamic type of *DMG(2)* by means of [VR], *DMG(2)*'s dynamic conjunction and dynamic disjunction are derivable from the basic operations of conjunction and disjunction, and quantifiers may take scope over further discourse. So combined dynamics and flexibility enables us to derive the dynamic interpretations of sentences in *DMG(2)* from the basic interpretations of the constituent expressions. And what is more, the system of type flexibility is structurally richer then *DMG(2)*. The following example may serve as an illustration of the real potential of dynamic flexibility:

(22) Every customer is offered coffee, that is, if he looks wealthy. He is sent up to me as quickly as possible. If he doesn't look wealthy, he can wait.

Examination of example 22 reveals that it exhibits complicated semantic dependencies. The main clause in this example is the conditional *B if A*. The conditionalization expressed by *if he looks wealthy*, syntactically applies to the sentence *Every customer is offered coffee*, while, semantically, it breaks in: the conditionalization takes scope over the clause *is offered coffee*, which is what is conditionalized, while it itself remains in the scope of the subject of the clause *every customer*, which binds the pronoun in the condition. Furthermore, this conditionalization induces a scope shuffle, which has the effect that the second sentence *He is sent up etc.* also lands in the scope of the conditional, whereas the third sentence *If he doesn't look wealthy etc.* does not land in the scope of the conditional (what it evidently should not do), while it does stay within the scope of the quantifying noun phrase *every customer*. These semantic dependencies can be accounted for with the type changing rules at hand. The following type transitions suffice:

$$[1\mathrm{AR}]([2\mathrm{VR}]([2\mathrm{VR}](\textit{is offered coffee}\,'))) =$$
$$\lambda T \lambda Q\ {}^\vee T(^\wedge \lambda x\ {}^\vee Q(^\wedge \lambda R\ {}^\vee R(^\wedge\ \mathtt{c\_off}({}^\vee x))))$$

The first value raising of the verb phrase creates the landing site R for conjunction with the second sentence in the example. The second value raising creates the landing site Q for the conditionalization with the *if*-clause. This is a raising *over* the first raising (a raising to the level of quantifiers over quantifiers over propositions), which creates room for the *if*-clause scope shuffle. Finally, the first argument is raised in order for the subject *every customer* to take wide scope. Application of this translation of the verb phrase to that of the subject gives the following translation:

$$\lambda Q\ \forall d(\mathtt{cust}(d) \rightarrow {}^{\vee}Q({}^{\wedge}\lambda R\ {}^{\vee}R({}^{\wedge}\ \mathtt{c\_off}(d))))$$

The *if*-clause is treated as follows:

$$[1\mathrm{AR}]([3\mathrm{VR}]([1\mathrm{GD}\text{-}_{T}](\textit{if he looks wealthy}\,'))) =$$
$$\lambda O \lambda R \lambda S\ {}^{\vee}O({}^{\wedge}\lambda P\ {}^{\vee}S({}^{\wedge}\ \mathtt{wealthy}(d) \rightarrow {}^{\vee}P(R)))$$

The division of the first argument makes the landing site for the second sentence *inside* of the first argument of the *if*-clause accessible: the second argument of the clause is transmitted to its first argument. The raising of the third value creates a landing site for the conjunction with the third sentence in our example. This conjunction, therefore, takes scope over the implication itself. Finally, the raising of the first argument effects that the whole sentence to which the *if*-clause applies takes scope over the implication, and over the conjunction with the third sentence, and that the dynamic implication applies to material within the translation of its syntactic argument. Application of the *if*-clause to (the intension of) its argument yields the following translation of the first sentence of 22:

$$\lambda R \lambda S\ \forall d(\mathtt{cust}(d) \rightarrow {}^{\vee}S({}^{\wedge}\ \mathtt{wealthy}(d) \rightarrow {}^{\vee}R({}^{\wedge}\ \mathtt{c\_off}(d))))$$

Finally, this translation applies to the intension of conjunction with the second sentence and with that of the third:

$$\lambda p\ ({}^{\vee}p \wedge \mathtt{sent\_up}(d))$$
$$\lambda p\ ({}^{\vee}p \wedge (\neg\mathtt{wealthy}(d) \rightarrow \mathtt{wait}(d)))$$

which produces the right interpretation of example 22:

$$\begin{aligned}\forall d(\mathtt{cust}(d) \rightarrow\ & ((\mathtt{wealthy}(d) \rightarrow (\mathtt{c\_off}(d)\ \wedge \mathtt{sent\_up}(d))) \wedge \\ & (\neg\mathtt{wealthy}(d) \rightarrow \mathtt{wait}(d))))\end{aligned}$$

The last example may have convinced the reader of the strenghth of combined dynamics and flexibility in a compositional theory of interpretation. As is to be expected, however, there is a reverse to this. We have seen that Hendriks' version of Montague grammar with added type flexibility derives all the quantifier scope configurations that respect the application structure of a sentence. Now that we have extended the grammar to a (rudimentary) discourse grammar, the same holds for the possible scopings of the quantifiers in a complete *text*. Clearly, we have a source of overgeneration here.

Consider, for instance, the case where a quantifier occurs in a sentence in the middle of a stretch of text. It is easily shown that in our system the quantifier may take scope over the entire part of text. This, we think, should not be. We do want to allow for the possibility that a quantifying noun phrase takes scope over an arbitrary part of text *after* (to the right of) the noun phrase's occurrence. Notably, indefinite noun phrases have this potential. On the other hand, we do not want that our system generates cataphoric relationships extending over an arbitrary length of text. But, as it stands, noun phrases in our system do have this possibility of unlimited backwards binding. (Of course, we want to account for cataphoric relationships, and we can do so. What we do not want is that these are generated without restriction.). And even something stronger holds. Like we said, with the rule of division expressions of type $\langle {}^{\frown}t, t\rangle$ may climb up in an application structure. But this means that, for instance, a conditionalization *If A then ...*, translated as

$\lambda q\ A' \rightarrow {}^{\vee}q$, not only may come to range over further discourse, but over preceding discourse as well, and that is clearly ridiculous.

The point to be made here, is that we can exclude, in a structural way, the possibility that pronouns get bound by quantifying noun phrases occurring in successive parts of text. For instance, if we only allow [1AR] and [GD] to operate on the sequencing operator, we exclude that a quantifying noun phrase takes scope over the sequencing operator to the left of it, and, thus, we exclude backwards binding while preserving the possibility of forwards binding. However, we have to be careful here. Restrictions on the application of type changes to certain expressions or operations might be easily circumvented in a flexible system. Therefore, an effective restriction on type change will be twofold. First, the whole system of type change will be located in the lexicon. All type changes operate on lexical atoms only, and not on compound expressions anymore. Second, any restrictions on the application of type changing rules are stored, in the lexicon, with the relevant atoms. So, in the case at hand, the proposal would be that the sequencing operator is stored, in the lexicon, with the restriction that [VR] is not allowed to apply to it, nor applications of [nAR] where $n > 1$.

We will not discuss the restricion on backwards binding further here. Alternative, more liberal or more severe, ways of forcing the sequencing operation to be directional are conceivable, but we postpone discussion about the whole issue of restrictions on type change to some other occassion. Here, it suffices to note that effective management of the number of derivable readings within a flexible grammar is possible. Another issue is more urgent now, which is the treatment of wide scope downward monotonic expressions in a flexible dynamic Montague grammar. This is the topic of the next section.

3.3 Type change and downward monotonicity

How are we to derive the right dynamic translations of negative operators by means of type flexibility? If we just let the type changing operations apply, we, again, face the problems we had in *DMG* and *DMG(2)* with dynamic negation as mere complementation. However, also in the flexible system a solution is possible. In this section we propose a logical modification of the interpretation of type changes that is to be used when downward monotonic expressions are subjected to type change. Some examples next show that the derived translations of downward monotonic expressions gives us all of the *DMG(2)*-readings discussed above, and, furthermore, the possibility to generalize the *DMG(2)* treatment to even more puzzling and complex examples.

As is to be expected on the basis of the discussion on dynamic negation in the sections 1 and 3, it is awkward to extend the scope of downward monotonic expressions without further ado. Therefore, we will provide the type shifts that regulate the scope of such expressions with an amended interpretation, like we did with the dynamic (wide scope) negation in the system of *DMG(2)*. Before we state the amended rules, we explicitly define a general notion of monotonicity which fits the generalized format of type change:

Definition 3.8 (Monotonicity)

1. If ϕ is an expression of type $\langle\vec{a}^{n-1}, \langle\langle\vec{b}^m, t\rangle, \langle\vec{c}^j, t\rangle\rangle\rangle$, x_i, y_i, and p and q metavariables of type a_i, c_i, and $\langle\vec{b}^m, t\rangle$, then ϕ is upward monotonic in its n-th argument iff
$\forall_{M,s,g}\forall_{\vec{x}^{n-1}}\forall_{p,q}\forall_{\vec{y}^j}(p \subseteq q \rightarrow ([\![\phi]\!]_{M,s,g}(\vec{x}^{n-1})(p)(\vec{y}^j) \rightarrow [\![\phi]\!]_{M,s,g}(\vec{x}^{n-1})(q)(\vec{y}^j)))$
2. If ϕ is an expression of type $\langle\vec{a}^{n-1}, \langle\langle\vec{b}^m, t\rangle, \langle\vec{c}^j, t\rangle\rangle\rangle$, x_i, y_i, and p and q metavariables of type a_i, c_i, and $\langle\vec{b}^m, t\rangle$, then ϕ is downward monotonic in its n-th argument iff

$$\forall_{M,s,g}\forall_{\vec{x}^{n-1}}\forall_{p,q}\forall_{\vec{y}^j}(p \supseteq q \rightarrow ([\![\phi]\!]_{M,s,g}(\vec{x}^{n-1})(p)(\vec{y}^j) \rightarrow [\![\phi]\!]_{M,s,g}(\vec{x}^{n-1})(q)(\vec{y}^j)))$$

Although it may look a bit cumbersome, this definition of up- and downward monotonicity is a mere generalization of the standard one. Inspection of our fragment reveals that it contains four downward monotonic expressions. These are *every*, which is downward monotonic in its 1-st argument, *no*, downward monotonic in its 1-st and 2-nd argument, and *not* and *if*, both downward monotonic in their 1-st argument. Now, when we subject these operators to type change, with associated interpretation as defined in 3.4, the monotonicity properties of these operators, and of other operators are disturbed. For instance, if we raise (over) the n-th argument of an expression in which it is downward monotonic, it gets upwards monotonic there. On the other hand, an upward monotonic operator may be turned into a downward monotonic operator, if one of its argument expressions is downward monotonic and takes wide scope over the functor. In what follows, we will associate adjusted interpretations with the type shifts which are to secure that the operators in our fragment preserve their proper monotonicity properties. Hereby we hold on to the strategy followed in section 2, where we defined dynamic negation. The basic idea is again to dualize the interpretation of the extended scope of downward monotonic operators, and, conversely, to dualize the interpretation of expressions that escape from the scope of a downward monotonic operator by raising. More specifically, if we divide the argument of an operator in which it is downward monotonic, the divided operator will take the dual of the interpretation of the argument divided into, since this argument does not belong to the syntactic scope of the operator; and if we raise (over) an argument in which an operator is downward monotonic, then the raised argument is enclosed by two negation signs, in order to preserve the monotonicity properties of both the argument and the functor. However, before we give the definition of dual type change, we must define a generalized dual and a generalized negation, since the type changes and the intended duals act on all types. The notions of generalized negation $\sim$ and generalized dual * are defined as follows:

Definition 3.9 (Generalized dual negation)

1. $\sim\phi_e = \phi = \phi_e{}^*$
2. $\sim\phi_t = \neg\phi$; $\phi_t{}^* = \phi$
3. $\sim\phi_{\langle a,b\rangle} = \lambda x_a \sim\phi(x^*)$; $\phi_{\langle a,b\rangle}{}^* = \lambda x_a \sim\phi(\sim x)$
4. $\sim\phi_{\langle s,a\rangle} = {}^\wedge\!\sim{}^\vee\phi$; $\phi_{\langle s,a\rangle}{}^* = {}^\wedge({}^\vee\phi)^*$

A simple induction suffices to prove the law of double generalized negation, and that of the double generalized dual:

Fact 3.4

1. $R^{**} = R$
2. $\sim\sim\Phi = \Phi$

We must point out here that the definition of generalized negation does not correspond to the interpretation set of natural language negation. There is a large overlap between the two, but both kinds of negation serve different purposes: the generalized negation serves a technical purpose, the interpretation of natural language negation a semantic one.

With the remarks made above in mind we may now turn to the definition of the amended interpretations of the type shifts of downward monotonic expressions:

Definition 3.10 (Dual type changes)

[nDVR] If ϕ is an expression of type $\langle\vec{a}^{n-1}, b\rangle$ downward monotonic in its n-th argument, Y a variable of type $^\frown\langle^\frown b, t\rangle$, $\vec{x}^{n-1}$, Y and y are not free in ϕ, then

$$\phi \Longrightarrow \lambda\vec{x}^{n-1}\lambda Y \ {\sim}^{\vee}Y(^{\wedge}\lambda y \ {\sim}\phi(\vec{x}^{n-1})(y))$$

[nDAR] If ϕ is expression of type $\langle\vec{a}^{n-1}, \langle b, \langle\vec{c}^m, t\rangle\rangle\rangle$ downward monotonic in its n-th argument, y and Y variables of type b and $^\frown\langle^\frown\langle b, t\rangle, t\rangle$ respectively, $\vec{x}^{n-1}$, Y, $\vec{z}^m$ and y are not free in ϕ, then

$$\phi \Longrightarrow \lambda\vec{x}^{n-1}\lambda Y\lambda\vec{z}^m \ {\sim}^{\vee}Y(^{\wedge}\lambda y \ {\sim}\phi(\vec{x}^{n-1})(y)(\vec{z}^m))$$

[nDGD(c)] If ϕ is an expression of type $\langle\vec{a}^{n-1},\langle^\frown\langle\vec{b}^m, \langle\vec{d}^j, t\rangle\rangle,\langle\vec{d}^j, t\rangle\rangle\rangle$ downward monotonic in its n-th argument, z and Y variables of type c and $^\frown\langle\vec{b}^m, \langle c, \langle\vec{d}^j, t\rangle\rangle\rangle$ respectively, $\vec{x}^{n-1}$, Y and z not free in ϕ, then

$$\phi \Longrightarrow \lambda\vec{x}^{n-1}\lambda Y\lambda z \ \phi(\vec{x}^{n-1})(^{\wedge}\lambda\vec{y}^m \ {}^{\vee}Y(\vec{y}^m)(z^*))$$

With the dual type changes, we can now also derive the two remaining operators of *DMG(2)*, dynamic negation and dynamic implication, from basic negation and implication, as appears from the following fact (again, R is a variable of type $\langle s, \tau\rangle$, p of type $\langle s, t\rangle$, and Φ and Ψ are expressions of type $\langle\langle s, \tau\rangle, t\rangle$):

Fact 3.5

1. ${\sim}^{\ddagger}\Phi = \lambda R\ \neg\Phi(R^*) = [\text{1DGD}](\textit{not}\,')(^{\wedge}\Phi)$
2. $\Phi \Rightarrow \Psi = \lambda R\ \neg\Phi(^{\wedge}\lambda p\ {}^{\vee}p \wedge \neg\Psi(R)) = [\text{1DAR}]([\text{2GD}](\textit{if}\,'))(^{\wedge}\Phi)(^{\wedge}\Psi)$

So, in *FDMG*, the *DMG(2)*-interpretations of dynamic sentential operators are all derivable from basic interpretations of their static counterparts. On the other hand, we still cannot completely derive the *DMG(2)* interpretations of the determiners from their static counterparts, since a raised first argument takes scope over the determiner itself. But, notice that the raising of the first argument of the determiner *every* now induces a dual interpretation, since the determiner is downward monotonic in that argument. Therefore, the (dynamic) *FDMG* interpretation of the donkey sentence 3, which has a dynamic common noun phrase, equals its interpretation in the dynamic Montague grammars and in *DRT*.

We are now in a position to address the issue of [AR]-elimination. Above we argued that applications of [AR] to a function could be eliminated in favour of an application of [VR] to the function *and* an application of [GD] on the relevant argument of the function. This is no longer true in the revised system that makes use of dual type changes. To be precise, if [iAR] applies to a function whose i-th argument is a downward monotonic quantifier, then the elimination of [AR] does not preserve truth-conditional content since the required application of [GD] to the argument would get its dual interpretation. Therefore, it makes a difference now if we choose to adopt [AR], or to eliminate it from the type changing system. We choose to eliminate it, for the following reason.

If we use a system with [AR], we can assign a downward monotonic expression like *no man* wide scope over continuations of discourse by applying [AR] (among others) to the functor, and without changing the type of the quantifying noun phrase itself. In that case no dual type changes would be involved, and the extension of the scope of the negative operator would not be dualized. On the other hand, if we exclude [AR] from the system of type change, which is what we propose, then we can only extend the scope of quantifiers by changing their type (and that of the function to which they apply). In other words, if [AR] is replaced by [GD], then the extended scope of a downward monotonic expression will be paired to a dualized interpretation. So this is the reason for favouring [GD] at the cost of [AR]. However, in all the cases where [AR] accomodates a function to an upward monotonic quantifying argument, the application of [AR] remains eliminable, and therefore we can continue using such instances of [AR] in the sequel for ease of exposition. And, furthermore, applications of [DAR] that accomodate a function to an upward monotonic daughter can be eliminated as well. The following fact shows the elimination of [1DAR]:

Fact 3.6 ([DAR] elimination) If α is not 1-downward monotonic, then

$$\begin{gathered}([1\mathrm{DAR}](\beta_{\langle a,\langle \vec{b}^n,t\rangle\rangle}))({}^\wedge\alpha_{\langle {}^\wedge\langle a,t\rangle,t\rangle}) = \lambda\vec{y}^n\ \neg\alpha({}^\wedge\lambda x\ \neg\beta(x)(\vec{y}^n)) = \\ (\lambda T\lambda\vec{y}^n\ \neg{}^\vee T({}^\wedge\lambda x\lambda\vec{y}^n\ \neg\beta(x)(\vec{y}^n{}^*))(\vec{y}^n{}^*))({}^\wedge\lambda P\lambda\vec{y}^n\ \alpha({}^\wedge\lambda x\ {}^\vee P(x)(\vec{y}^n))) \\ = ([1\mathrm{DVR}_{\langle \vec{b}^n,t\rangle}](\beta))({}^\wedge[1\mathrm{GD}_{\vec{b}^n}](\alpha))\end{gathered}$$

So we continue using instances of [AR] and [DAR] in the sequel, but only if these make a function applicable to an upward monotonic argument.

We will now illustrate the system of flexible dynamic Montague grammar by discussing the derivations of readings assigned to some examples discussed above. We reconsider three examples in which downward monotonic expressions take wide scope. In the appendix we show how all the other examples that are treated in sections 1 and 2, are assigned equivalent interpretations within the framework of *FDMG*. The first example is example 4, repeated as 23:

(23) It is not the case that John owns no car. It is standing in front of the house.

First we lift the transitive verb *own* into the dynamic format by means of [3VR]. Next, we must make it applicable to the quantifying noun phrase *no car*. However, we shouldn't use [AR] now, because then we would assign the downward monotonic noun phrase *no car* wide scope without using a dualized interpretation. Therefore, in order to let the transitive verb apply to the quantifying noun phrase, we raise the verb over its first argument, and divide the quantifying noun phrase to which it applies. Since the quantifying noun phrase is downward monotonic, two duals are introduced. The translation of the verb phrase *owns no car* now reads:

$$\begin{gathered}([1\mathrm{VR}_{\langle {}^\wedge e,\langle {}^\wedge\tau,t\rangle\rangle}]([3\mathrm{VR}](\mathit{own}')))({}^\wedge\ [1\mathrm{DGD}_{{}^\wedge e,{}^\wedge\tau}](\mathit{no\ car}')) = \\ \lambda x\lambda R\ \neg\exists d(\mathtt{car}(d) \wedge {}^\vee R^*({}^\wedge\ \mathtt{own}(d)({}^\vee x^*)))\end{gathered}$$

Application of this expression to the intension of the translation of *John*, the dual of which is *John* itself, yields the following translation:

$$\lambda R\ \neg\exists d(\mathtt{car}(d) \wedge {}^\vee R^*({}^\wedge\ \mathtt{own}(d)(\mathtt{john})))$$

The sentence negation *it is not the case that* must be divided, and we use the dynamic interpretation of the division, $\lambda P\lambda R\ \neg{}^\vee P(R^*)$. If we apply this translation to the intension of the translation of *John owns no car*, we get the translation of the first sentence of 23:

$$\begin{gathered}\lambda R\ \neg\neg\exists d(\mathtt{car}(d) \wedge {}^\vee R^{**}({}^\wedge\ \mathtt{own}(d)(\mathtt{john}))) = \\ \lambda R\ \exists d(\mathtt{car}(d) \wedge {}^\vee R({}^\wedge\ \mathtt{own}(d)(\mathtt{john})))\end{gathered}$$

This, in fact, is the dynamic translation of the sentence *John owns a car*, and a straightforward conjunction with the second sentence of 23 produces the correct reading of the example:

$$\exists d(\mathtt{car}(d) \wedge \mathtt{own}(d)(\mathtt{john}) \wedge \mathtt{before\_the\_house}(d))$$

The second example is 10, repeated as 24:

(24) No player leaves. He stays.

Again, the verb in the first sentence is raised to the dynamic type, and we raise it over the first value in order to accomodate it to its quantifying argument expression. We divide the quantifying subject argument as well, and we use the dual interpretation of the division:

$$\begin{gathered}([1\mathrm{VR}_{\langle{}^{\cdot}\mathcal{T},t\rangle}]([2\mathrm{VR}](\textit{leave}')))(^\wedge\ [1\mathrm{DGD}_{{}^{\cdot}\mathcal{T}}](\textit{no player}')) = \\ \lambda R\ \neg\exists d(\mathtt{player}(d) \wedge {}^\vee R^*(^\wedge\ \mathtt{leave}(d))) = \\ \lambda R\ \forall d(\mathtt{player}(d) \rightarrow {}^\vee R(^\wedge\ \neg\mathtt{leave}(d)))\end{gathered}$$

In fact, this is also the dynamic translation of *Every player does not leave.* A straightforward conjunction with *He stays.* yields the correct translation of 24:

$$\forall d(\mathtt{player}(d) \rightarrow (\neg\mathtt{leave}(d) \wedge \mathtt{stay}(d)))$$

The last example that we discuss here is 14, repeated as 25:

(25) If it is not the case that there is a bathroom downstairs, then it is upstairs.

In the antecedent of this conditional, a divided negation is applied to a dynamic existentially quantified sentence:

$$\begin{gathered}([1\mathrm{DGD}_{{}^{\cdot}\mathcal{T}}](\lambda p\ \neg{}^\vee p))(^\wedge\ \lambda R\ \exists d(\mathtt{bathroom}(d) \wedge {}^\vee R(^\wedge\ \mathtt{down}(d)))) = \\ \lambda R\ \neg\exists d(\mathtt{bathroom}(d) \wedge {}^\vee R^*(^\wedge\ \mathtt{down}(d)))\end{gathered}$$

We raise the implication operator in its first argument, and we must use the dual interpretation (notice that this is an eliminable instance of [DAR]):

$$[1\mathrm{DAR}](\lambda p\lambda q\ ({}^\vee p \rightarrow {}^\vee q)) = \lambda P\lambda q\ \neg{}^\vee P(^\wedge\ \lambda p\ ({}^\vee p \wedge \neg{}^\vee q))$$

Application to the intensions of the translation of the antecedent and that of the consequent $\mathtt{up}(d)$, gives the following interpretation of 25:

$$\begin{gathered}\neg(\lambda R\ \neg\exists d(\mathtt{bathroom}(d) \wedge {}^\vee R^*(^\wedge\ \mathtt{down}(d))))(^\wedge\lambda p\ ({}^\vee p \wedge \neg\mathtt{up}(d))) = \\ \exists d(\mathtt{bathroom}(d) \wedge \neg(\neg\mathtt{down}(d) \wedge \neg\mathtt{up}(d))) = \\ \exists d(\mathtt{bathroom}(d) \wedge (\mathtt{down}(d) \vee \mathtt{up}(d)))\end{gathered}$$

So we see that we can obtain *DMG(2)* results within the system of *FDMG*. In the adapted system of type flexibility, the characteristic *DMG(2)* interpretations of raised formulas, of dynamic negation, conjunction, disjunction and implication are all derivable from their basic interpretations. But, the system of type flexibility allows us to treat more intricate scope phenomena than the rigid dynamic Montague grammars do. We now show the *FDMG* treatment of two more interesting examples that are beyond the scope of *DMG* and *DMG(2)*. In the next section, some more examples are discussed.

First, consider example 26:

(26) If a farmer owns a donkey, he stones it, and if he leases, it he pleases it.

As in the donkey implications discussed above, the indefinite noun phrases *a farmer* and *a donkey* get universal import when raised out of the antecedent of the conditional, and they may bind pronouns in successive sentences. However, in this example the second sentence must not be conjoined with the consequent *he beats it* of the first implication, but with the whole implication itself, within the scope of both quantifying noun phrases. With our type changing system we can account for that. First the value of the transitive verb *own* must be raised, and next both of its original argument places are raised in order to apply to the two quantifying noun phrase:

$$([2\text{AR}]([1\text{AR}]([3\text{VR}](\textit{own}\,')))) (^\wedge\ \textit{a donkey}\,')(^\wedge\ \textit{a farmer}\,') =$$
$$\lambda R\ \exists d_i(\mathtt{farmer}(d_i) \wedge \exists d_j(\mathtt{donkey}(d_j) \wedge {}^\vee R(^\wedge\ \mathtt{own}(d_j)(d_i))))$$

Also the value of the implication operator is raised, and next its first argument is raised by means of [DAR]:

$$[1\text{DAR}]([3\text{VR}](\textit{if}\,')) =$$
$$\lambda P \lambda q \lambda R\ \neg{}^\vee P(^\wedge\ \lambda p\ \neg{}^\vee R(^\wedge\ {}^\vee p \rightarrow {}^\vee q))$$

We apply the resulting translation to the intension of the raised interpretation of the antecedent and to the intension of the consequent of the first implication $\mathtt{stone}(d_j)(d_i)$, and the first sentence reads after reduction:

$$\lambda R\ \neg\exists d_i(\mathtt{farmer}(d_i) \wedge \exists d_j(\mathtt{donkey}(d_j) \wedge \neg{}^\vee R(^\wedge\ \mathtt{own}(d_j)(d_i) \rightarrow \mathtt{stone}(d_j)(d_i)))) =$$
$$\lambda R\ \forall d_i(\mathtt{farmer}(d_i) \rightarrow \forall d_j(\mathtt{donkey}(d_j) \rightarrow {}^\vee R(^\wedge\ \mathtt{own}(d_j)(d_i) \rightarrow \mathtt{stone}(d_j)(d_i))))$$

We conjoin this translation with the intension of the translation of the second sentence $\mathtt{lease}(d_j)(d_i) \rightarrow \mathtt{please}(d_j)(d_i)$, and we get at the right reading of example 26:

$$\forall d_i(\mathtt{farmer}(d_i) \rightarrow \forall d_j(\mathtt{donkey}(d_j) \rightarrow \quad ((\mathtt{own}(d_j)(d_i) \rightarrow \mathtt{stone}(d_j)(d_i)) \wedge$$
$$(\mathtt{lease}(d_j)(d_i) \rightarrow \mathtt{please}(d_j)(d_i)))))$$

(27) No farmer beats a donkey, if he isn't insane. He will not yell at it either. If he is insane, then he might beat a donkey.

This example must be compared to example 22 which exhibits the same (intended) semantic structure. The main difference with example 22 is that the quantifying noun phrase that has widest scope is not the upward monotonic noun phrase *every customer*, but the downward monotonic noun phrase *no farmer*. However, this downward monotonic noun phrase should not be given wide scope over the sentence as such, but over its (generalized) dual. As the derivation below shows, the reading that results from assigning *no farmer* dual wide scope equals the reading of example 27, in which *no farmer beats* is replaced by *every farmer does not beat*, and where *every farmer* is given wide scope as in the treatment of example 22. The difference in the derivation of a reading of 27 and of 22 are due to the fact that we should not use instances of [AR] now, for reasons explained above.

First we raise the value of the verb phrase in example 27 three times:

$$[1\text{VR}_{\langle \vec{a}^3, t\rangle}]([2\text{VR}_{\langle \vec{b}^2, t\rangle}]([2\text{VR}](\textit{beats a donkey}\,'))) =$$
$$\lambda T_{\hat{}d} \lambda P_{\hat{}c} \lambda R_{\hat{}\tau} \lambda S_{\hat{}\tau}\, {}^\vee T(^\wedge \lambda x_{\hat{}e} \lambda Q_{\hat{}c}\, {}^\vee Q(^\wedge \lambda R_{\hat{}\tau}\, {}^\vee R(^\wedge \mathtt{BAD}({}^\vee x))))(P)(R)(S)$$

(where $\vec{a}^3 = \hat{}c, \vec{b}^2,\ \ \vec{b}^2 = \hat{}\tau, \hat{}\tau,\ \ c = \langle \hat{}\langle \hat{}\tau, t\rangle, \langle \vec{b}^2, t\rangle\rangle,\ d = \langle \hat{}\langle \hat{}e, \langle \vec{a}^3, t\rangle\rangle, \langle \vec{a}^3, t\rangle\rangle$, and where $\mathtt{BAD}$ abbreviates the simplest translation of *beats a donkey*.) As we required, the noun phrase *no farmer* can only take wide scope if it is subjected to [DGD]:

$$[1\text{DGD}_{\vec{a}^3}](\textit{no farmer}\,') =$$
$$\lambda Q_{\hat{}\langle \hat{}e, \langle \vec{a}^3, t\rangle\rangle} \lambda P_{\hat{}c} \lambda R_{\hat{}\tau} \lambda S_{\hat{}\tau} \neg\exists d(\mathtt{farmer}(d) \wedge {}^\vee Q(^\wedge d)(P^*)(R^*)(S^*))$$

Application of the first translation to the intension of the second one gives, with reduction:

$$\lambda P_{\hat{}c} \lambda R_{\hat{}\tau} \lambda S_{\hat{}\tau} \neg\exists d(\mathtt{farmer}(d) \wedge {}^\vee P^*(^\wedge \lambda R_{\hat{}\tau}\, {}^\vee R(^\wedge \mathtt{BAD}(d)))(R^*)(S^*)) =$$
$$\lambda P_{\hat{}c} \lambda R_{\hat{}\tau} \lambda S_{\hat{}\tau} \neg\exists d(\mathtt{farmer}(d) \wedge (\sim{}^\vee P^*(^\wedge \sim\lambda R_{\hat{}\tau}\, {}^\vee R(^\wedge \mathtt{BAD}(d))))(R^*)(S^*)) =$$
$$\lambda P_{\hat{}c} \lambda R_{\hat{}\tau} \lambda S_{\hat{}\tau} \forall d(\mathtt{farmer}(d) \rightarrow {}^\vee P(^\wedge \lambda R_{\hat{}\tau}\, {}^\vee R(^\wedge \neg\mathtt{BAD}(d)))(R)(S))$$

The following type changes are applied to the antecedent of the implication:

$$\begin{gathered}[1\mathrm{VR}_{\langle \vec{b}^2,t\rangle}]([3\mathrm{VR}]([1\mathrm{GD}_{^\frown\tau}](\textit{if he isn't insane}\,'))) = \\ \lambda T_{^\frown\langle \vec{a}^3,t\rangle} \lambda R_{^\frown\tau} \lambda S_{^\frown\tau} {}^\vee T({}^\wedge \lambda P_{^\frown\langle ^\frown\tau,t\rangle} \lambda R_{^\frown\tau} \lambda S_{^\frown\tau} {}^\vee S({}^\wedge\ \mathtt{sane}(d) \rightarrow {}^\vee P(R)))(R)(S)\end{gathered}$$

Application of this translation to the intension of the translation of *No farmer beats a donkey*, gives the following translation of the first sentence of example 27:

$$\begin{aligned} \lambda R_{^\frown\tau} \lambda S_{^\frown\tau} \forall d(\mathtt{farmer}(d) \rightarrow\ & ({}^\wedge \lambda P_{^\frown\langle ^\frown\tau,t\rangle} \lambda R_{^\frown\tau} \lambda S_{^\frown\tau} {}^\vee S({}^\wedge\ \mathtt{sane}(d) \rightarrow {}^\vee P(R))) \\ & ({}^\wedge \lambda R_{^\frown\tau} {}^\vee R({}^\wedge \neg \mathtt{BAD}(d)))(R)(S)) = \\ \lambda R_{^\frown\tau} \lambda S_{^\frown\tau} \forall d(\mathtt{farmer}(d) \rightarrow\ & {}^\vee S({}^\wedge\ \mathtt{sane}(d) \rightarrow {}^\vee R({}^\wedge \neg \mathtt{BAD}(d)))) \end{aligned}$$

This last translation applies to the intension of the translation of the second sentence *He will not yell at it either*, which reads:

$$\lambda p\ ({}^\vee p \wedge \neg \mathtt{yell\_at\_it}(d))$$

and next to that of the third sentence *If he is insane, then he might beat a donkey*, which reads:

$$\lambda p\ ({}^\vee p \wedge (\neg \mathtt{sane}(d) \rightarrow \mathtt{maybe\_BAD}(d)))$$

and the interpretation that results, finally, reads as follows:

$$\begin{aligned} \forall d(\mathtt{farmer}(d) \rightarrow\ & ((\mathtt{sane}(d) \rightarrow (\neg \mathtt{BAD}(d) \wedge \neg \mathtt{yell\_at\_it}(d))) \wedge \\ & (\neg \mathtt{sane}(d) \rightarrow \mathtt{maybe\_BAD}(d)))) \end{aligned}$$

(Notice that we can also derive an appropriate reading in which *a donkey* is assigned scope over the whole sequence. We leave the derivation of this reading to the interested reader.)

In order to conclude this section, we briefly indicate some other differences between the systems of *DMG(2)* and *FDMG*, and point out some inadequacies of the latter. We already indicated above that *DMG(2)* and *FDMG* largely overlap, but also that *FDMG* is much more expressive than *DMG(2)*. The treatment of the examples 22, 26 and 27 exemplifies its richess. (Moreover, in the next section we show the *FDMG* treatment of a few more examples that fall beyond the scope of *DMG(2)*.) On the other hand, the two systems differ with respect to their treatment of dynamic common noun phrases, and with respect to this issue, *DMG(2)* fares better than *FDMG* does. Like we said above, dynamic binding by quantifiers figuring in a common noun phrase can only be established in *FDMG* by raising these over the determiner head of the noun phrase. But this is not very satisfactory. For instance, if this is the way to treat donkey sentences with a universal determiner, we create our own *FDMG*-specific binding paradox. Consider the following example:

(28) Every farmer who beats a donkey that he owns also kicks it.

On its most likely reading(s), this sentence exhibits two anaphoric relationships: the pronoun *he* is bound in the restriction of the determiner *every*, and the pronoun *it* is bound by the quantifying noun phrase *a donkey that he owns*. In order to establish an anaphoric relationship between the noun phrase *a donkey that he owns* and the pronoun *it*, the noun phrase has to be raised over the determiner *every* in *FDMG*. But if we do so, then also the pronoun *he* is raised out of the scope of the determiner *every*, and it will end up unbound. A view on the application structure of example 28 next reveals that it is impossible to establish both anaphoric relationships in *FDMG*. So, example 28 presents a real problem to *FDMG*. We return to this issue in section 4.2.

A last thing to point out here is that the preservance of monotonicity properties in the process of type change is not waterproof. The dual interpretations of the type changes may have proved themselves by now. However, we may fail to detect negative operators for which dual interpretations of type change are required, since in certain cases their downward monotonicity can be embezzled by the very type changes themselves. We will not go into this problem here, though, because in view of considerations laid down in the next section, it may be advisable for the moment to await a more restricted and more manageable system of type change.

4 Further directions

4.1 The scope of negation within sentences

We already remarked that the transition from the dynamic Montague grammars defined in the sections 1 and 2 to the system of *FDMG* involves a shift in perspective on the phenomenon of dynamic binding. In the dynamic Montague grammars we account for intersentential anaphoric relationships by means of an inherently dynamic and directional notion of sequencing. In *FDMG*, on the other hand, these relationships are all instances of the phenomenon of scope and binding, which itself is non-directional. This shift in perspective also has a reverse. The dualized interpretation of wide scope negation, and of other downward monotonic expressions, is motivated by and large by our intuitions concerning extra-sentential binding by downward monotonic quantifiers in discourse. However, the dual interpretation is defined for all type changes of downward monotonic expressions, and therefore an extension of the scope of a downward monotonic expression within a sentence is also associated with a dual interpretation. So, with respect to the issue of wide scope downward monotonic expressions *within* sentences, our theory generates readings for sentences that differ from readings derived in Hendriks' system of type change. In this section we discuss three examples that seem to favour our analysis.

Remember that we proposed to force the sequencing operator to be directional in the sense that it does not allow for backwards binding of pronouns by quantifying noun phrases. However, we excluded only cataphoric relationships between terms that occur in two different sentences in a sequence of sentences. Still, at the sentential level no restrictions on the directionality of binding obtains as yet. Therefore we can account for cataphoric relationships within sentences. Consider the following type change of the implication operator:

$$[2\text{AR}](\mathit{if}') = \lambda p \lambda Q\ {}^{\vee}Q({}^{\wedge}\lambda q\ ({}^{\vee}p \rightarrow {}^{\vee}q))$$

On the basis of this translation we can account for the following example in which a cataphoric relationship occurs:

(29) If he is in danger, every man prays to God.

A correct interpretation of this sentence results if we apply the raised translation of *if* to the intension of the basic translation of *he is in danger*, $\mathbf{in\_danger}(d)$, and the intension of the dynamic translation of *every man prays to god*, $\lambda R\ \forall d(\mathtt{man}(d) \rightarrow {}^{\vee}R({}^{\wedge}\ \mathtt{pray\_to\_God}(d)))$:

$$\forall d(\mathtt{man}(d) \rightarrow (\mathbf{in\_danger}(d) \rightarrow \mathtt{pray\_to\_God}(d)))$$

Since the derivation of this interpretation comprises no dual type changes, a structurally similar interpretation is derivable in Hendriks' system (although the pronoun in the antecedent of the implication will turn out to be unbound if *IL*, instead of *DIL*, is the framework of interpretation). However, in the following example, which is much like example 29, we must assign wide scope to the downward monotonic quantifying noun phrase *no man*:

(30) If he isn't insane, no man beats a donkey.

Now, if we were to assign the noun phrase *no man* wide scope without using the dual interpretation of the extended scope, the following interpretation would be the result (again BAD abbreviates the basic translation of *beats a donkey*):

$$\neg\exists d(\mathtt{man}(d) \wedge (\neg\mathtt{insane}(d) \rightarrow \mathtt{BAD}(d)))$$

If one rephrases this formula for oneself, it sounds allright: *No man is such that, if he is not insane, he beats a donkey.* But we have to be cautious. The real truth-conditions of the latter formula are that no man is insane, and that no man beats a donkey. Clearly, this is not the interpretation of example 30. However, according to our proposal a different interpretation results from assigning the noun phrase *no man* wide scope over the implication. Since this noun phrase is downward monotonic, it is the dual of the conditionalization *if he isn't insane* that lands in the scope of the quantifier. The translation of the consequent reads:

$$\lambda R\ \neg\exists d(\mathtt{man}(d) \wedge {}^{\vee}R^{*}({}^{\wedge}\ \mathtt{BAD}(d)))$$

Since the dual of the translation of *if he isn't insane* reads $\lambda q\ (\neg\mathtt{insane}(d) \wedge {}^{\vee}q)$, the interpretation associated with example 30 in *FDMG* is the following:

$$\neg\exists d(\mathtt{man}(d) \wedge \neg\mathtt{insane}(d) \wedge \mathtt{BAD}(d))$$

In other words, sentence 30 turns out to mean that no man who is not insane beats a donkey, or, equivalently, that for every man it holds that if the man is not insane, then he does not beat a donkey. To our opinion, this is the most likely reading of the example.

In the next example we generate a wide scope object interpretation with a downward monotonic object term:

(31) Some/at_least_one farmer beats no donkey.

If in the interpretation of this sentence the noun phrase *no donkey* is given wide scope without using the dual interpretation then we get the following reading:

$$\neg\exists d_j(\mathtt{donkey}(d_j) \wedge \exists d_i(\mathtt{farmer}(d_i) \wedge \mathtt{beat}(d_j)(d_i)))$$

It is true that intuitions get obscure if we try to imagine a wide scope object reading for examples like 31. Still, I myself do not succeed in associating this interpretation with the example. On the other hand, if the dual interpretation of the required type change of *no donkey* is used, the following reading emerges:

$$\begin{gathered}\neg\exists d_j(\mathtt{donkey}(d_j) \wedge \forall d_i(\mathtt{farmer}(d_i) \rightarrow \mathtt{beat}(d_j)(d_i))) = \\ \forall d_j(\mathtt{donkey}(d_j) \rightarrow \exists d_i(\mathtt{farmer}(d_i) \wedge \neg\mathtt{beat}(d_j)(d_i)))\end{gathered}$$

This reading I can associate with example 31, and this reading comes out if the example (read with a pitch on *some* and on *no*) is followed by a specification: *Some farmer beats no donkey: its owner.*

The last example that we discuss here concerns a de re belief with a downward monotonic noun phrase:

(32) John believes that no man is sane.

If we just let the quantifying noun phrase *no man* range over the belief attributed to John, without using a dual interpretation of the relevant type change, then the following interpretation results:

$$\neg\exists d(\mathtt{man}(d) \wedge \mathtt{believe}({}^{\wedge}\ \mathtt{sane}(d))(\mathtt{john}))$$

(Notice that it is awkward to talk about a de re belief in the present case. No belief is attributed, let alone a belief concerning some res.) Again, our intuitions may hesitate when confronted with the question whether the derived interpretation really is a reading of example 32. I am inclined to judge it not to be a reading of the example. Still, if we assume that people's beliefs are consistent, the reading derived without dual type change, is entailed by the reading derived in the system of *FDMG* with dual type change. In *FDMG* the de re interpretation (better: the de rebus interpretation) reads as follows:

$$\neg\exists d(\mathtt{man}(d) \wedge \neg\mathtt{believe}(^\wedge \neg\mathtt{sane}(d))(\mathtt{john})) = \\ \forall d(\mathtt{man}(d) \rightarrow \mathtt{believe}(^\wedge \neg\mathtt{sane}(d))(\mathtt{john}))$$

Whatever one may think of the interpretation derived without using dual type changes, the interpretation derived with dual type change to my opinion does constitute a reading of example 32. If John thinks of every man in some domain of discussion "This guy is insane", I think we might very well describe the situation by means of sentence 32. Clearly, the situation does not license one to conclude that John believes (de dicto) that all man are insane, but neither does that follow from the interpretation that we derived lastly.

We conclude that the examples 30–32 favour the use of dual interpretations of type changes that are required in order to handle intra-sentential scope phenomena, this in spite of the indeterminacy of our intuitions concerning (some of) these examples. What these examples at least show, is that the interpretations derived without dual type changes of downward monotonic expressions do not fare better then the interpretations derived in our system that uses dual type change. Of course, the behaviour of downward monotonic expressions at the sentential level constitutes a topic of its own, and one that has been studied extensively already, but limited space does not allow us to delve deeper into this topic here.

4.2 Conservativity and dynamic interpretation

If we resume the development of a dynamic Montague grammar, we come to a remarkable conclusion. The roots of the dynamic enterprise can be traced back to *DRT*. Groenendijk and Stokhof next reformulated *DRT* into *DPL*, and *DPL* again into the stronger system of*DMG*. We took *DMG* as the starting point of the present paper, and argued for the even richer system of *DMG(2)*, and from there it was a relatively small step into the formulation of the system of *FDMG*. However, there is something curious about this development. The first two of these five systems can be called genuinely dynamic in an intuitive sense. The interpretation of formulas in *DRT* and *DPL* is explained in terms of the changes that these formulas bring about in states of information. On the other hand, the resulting system of *FDMG* can hardly be called dynamic in this sense. In *FDMG*, dynamic binding turns up as a matter of the scope of quantifiers and operators, and intuitions concerning the left to right character of dynamic interpretation can be captured only by imposing a directionality constraints on top of the system of interpretation. In other words, the seemingly gradual development from *DRT* to *FDMG* has led to a crucial change in the approach to anaphora.

It need not come as a surprise that an account of scope phenomena can be used to deal with complex anaphoric relationships, since the issue of scope is clearly involved in dynamic binding. However, some things remain counterintuitive in the present approach. For instance, the flexible treatment of the donkey sentence *Every farmer who owns a donkey beats it* violates the commonly accepted restriction that relative clauses are scope islands. In the flexible system, this sentence can be assigned the right interpretation only by raising the noun phrase *a donkey* out of the relative clause in which it occurs. Now, if the relative clause constitutes a scope island, then this kind of raising is forbidden. Furthermore, as we indicated above, this approach to the donkey sentence gives way to *FDMG*'s scope paradox laid down in example 28.

Possibly related to this issue is a problem observed at the end of section 2. In the extended version of *DMG*, in *DMG(2)* and in *FDMG*, a dynamic universal quantifier gets existential force when it occurs in the antecedent of an implication or in the restriction on a universal quantifier and, likewise, if a dynamic implication occurs in the antecedent of an implication or in the restriction on a universal quantifier, the antecedent of the embedded implication turns out to be asserted. Clearly, the predicted readings of the corresponding sentences are not confirmed by linguistic fact. On the contrary, there seem to be no examples whatsoever of dynamic binding by a universal quantifier or an implication from within the restriction on a universal quantifier or from within the antecedent of an implication. Therefore, within the system of *FDMG* we somehow have to exclude the specific raisings that underlie these problematic readings, or we must try to find alternative interpretations of the type changes that do not generate these unwanted readings.

One way out of these problems might be to adopt (suitable adaptations) of Chierchia's conservative reformulation of *DMG*-operators ([6] and [7]). Chierchia joins the ranks of semanticists that have casted doubts on the prototypicality of the donkey sentences in which an existential quantifier turns out to have universal force (see, for instance Pelletier and Schubert [27]). In many cases, an existential quantifier in the antecedent of an implication binds pronouns in the consequent while it retains existential force. The following example poses a prototypical counterexample to the proposed universal reading of an existential quantifier:

(33) If I have a dime in my pocket, then I will throw it in the parking meter.

It needs no argumentation that anyone who sincerely utters 33 does not commit himself to throw all the dimes he has in his pocket in the parking meter. But that is the interpretation assigned to example 33 in *DRT* and in all the dynamic Montague grammars that we discussed. Here, we will not attempt to decide which kind of interpretation of such sentences is to be favoured. We just take it that the existential quantifier in these examples may be assigned universal as well as existential force. How to choose between the two readings remains obscure as yet, but world knowledge or lexical information might be involved as appears from a simple modification of example 33:

(34) If I have a dime in my pocket, then I will not throw it in the parking meter.

Clearly, example 34 does not imply that if I have some dimes in my pocket, then I will keep at least one for myself, which would be its interpretation if the existential quantifier is read existential. We will not go into further examples here (for further discussion, see also Heim [14]), but relate Chierchia's treatment of donkey implications with dime readings to the problems indicated above.

Chierchia departs from Groenendijk and Stokhof's *DMG*, and discusses the issue of the conservativity of dynamic determiners. In the theory of generalized quantifiers, conservativity is a logical property of (static) determiners D, defined as $D'(P)(Q) = D'(P)(P \cap Q)$. Intuitively, this says that the restriction on a determiner restricts the domain of evaluation of its second argument. For instance, the conservativity of the determiner *every* validates the equivalence between *Every man walks* and *Every man is a man who walks*. Chierchia observes that the dynamic determiner *every* in *DMG* is not conservative, and proposes a general conservative reformulation of the interpretation of all determiners (and of related adverbs of quantification). Chierchia's proposal comes down to applying determiners first to the closure of their first argument, and next to the *sequential composition* of the first argument with the second. For instance, if D is a determiner with static interpretation D', then its proposed dynamic interpretation reads:

$$\lambda P \lambda Q \lambda p \; D'(^\wedge\lambda x \downarrow{}^\vee P(x))(^\wedge\lambda x \; [{}^\vee P(x) \, ; {}^\vee Q(x)](p))$$

(In fact, Chierchia's proposal is a bit more involved, but for the present discussion this simplification suffices.) In this definition of dynamic determiners, special use is made both of the truth-conditional content of an expression and its dynamic import. The derived determiner first takes the closure of the dynamic interpretation of its first argument, and next this dynamic interpretation serves as the context of interpretation for the second argument. On this proposal, determiners remain conservative when transposed to the dynamic type. In other words, the donkey sentence *Every farmer who owns a donkey beats it* turns out equivalent with the sentence *Every farmer who owns a donkey is a farmer who owns a donkey and who beats it*. So, the donkey sentence is assigned the dime interpretation, with existential force of the noun phrase *a donkey*. An important advantage of Chierchia's proposal is that it easily accounts for asymmetric quantification by means of the determiner *most* and the related adverb of quantification *usually*, an issue that constitutes a problem in the framework of *DRT*. (For discussion, see Kadmon [19], Heim [14] and Chierchia [7].)

What is of interest to us here, is that Chierchia's proposal circumvents both problems sketched above. If we use the conservative dynamic interpretation of the universal quantifier (and a similar interpretation of the implication), the restrictive term (or the antecedent in the case of the implication) is frozen, and an embedded universal quantifier will not escape from there and get existential force. On the other hand, Chierchia only generates the dime interpretation of the donkey sentence, and not the original *DRT* interpretation with a universal interpretation of the embedded existential quantifier. However, the tools developed in this paper may serve to generate this other (non-conservative!) reading as well. To be precise, if we also allow the following dynamic interpretation of determiners that are downward monotonic in their first argument, the universal reading of the donkey sentence results again:

$$\lambda P \lambda Q \lambda p \; D'(^\wedge\lambda x \downarrow{}^\vee P(x))(^\wedge\lambda x \; [({}^\vee P(x))^* \, ; {}^\vee Q(x)](p))$$

Notice that the only difference with Chierchia's proposal is that this time it is the dual of the restrictive term that serves as the context of interpretation for the determiner's second argument.

We conclude that further investigations into the issue of dynamic conservativity may help to solve the problems indicated above. We will leave this subject here now, since it requires a lot more elaboration. However, we hope to have shown that the problems addressed at the beginning of the present section can be solved in some way or other. And it may very well be possible that this last, provisional, proposal initiates the development of a modified system of type change for dynamic interpretation, based on Chierchia's conservative dynamic lift and on the dual.

4.3 Conclusions

In this paper, we started discussing Groenendijk and Stokhof's *DMG*, developed to net *DRT*-results within the framework of traditional Montague grammar. We observed that *DMG* is easily modified into a system that gives a motivated dynamic interpretation of other logical operators than the existential quantifier, but with the exception of the crucial operation of negation. Next we turned to the system of *DMG(2)*, a dynamic Montague grammar typed one level higher (and equivalent to) *DMG* proper, that does allow the definition of an adequate notion of dynamic negation. Crucial for an adequate notion of dynamic negation, we argued, is that it applies to the dual of its extended scope. Next we showed that the system of *DMG(2)* (almost) appears as a subsystem of *FDMG* which consists of a simplified version of Montague grammar, phrased in terms of *DIL* and with added type flexibility. Again, we needed to amend the interpretation of type shifts of downward monotonic expressions in such a way that these expressions apply to the dual of the interpretation of their extended scope, and that expressions raised out of a downward monotonic context are surrounded by two negations, all of this in order to guarantee preservance of monotonicity properties. The resulting system of *FDMG*, we showed, enabled us to give an account of all kinds of complex and puzzling anaphoric relationships.

On the other hand, our treatment(s) of the dynamics of discourse raises further questions. For one thing, an extension of our fragment with other kinds of quantifying noun phrases is not without problems. For instance, it just would not be adequate to let non-monotonic noun phrases take scope over further discourse. An example is the noun phrase *exactly n men*. Further research is clearly needed here, research into the precise data concerning (the dynamics of) such noun phrases (for plurals, see van de Berg [5]), as well as into the possibility to account for these data by means of type flexibility.

Another question concerns the precise, and effective, formulation of restrictions on type change. One may think of restrictions excluding (certain cases of) backwards binding, restrictions securing the use of the proper (dual or not dual) type changing operations, and other possible constraints on derivable scope configurations. In view of this question, we need not exclude beforehand that an effective management of the system of type change requires an adjusted formulation of the type changing rules themselves.

A modification of the system of type change may also be required in view of the phenomena discussed in the preceding section. Notably, *FDMG*'s scope paradox (example 28) poses a serious challenge to the present undertaking, and so do the examples that seem to involve a conservative interpretation of certain operators. Here too, further comprehensive research must help in settling the matter. What is most important, however, is that we hope to have shown what kinds of techniques allow us to deal with the extended scope of negation and of negative operators in discourse.

5 Appendix

We turn to the proof of some facts stated in this paper. Before we turn to the facts themselves, we prove some usefull lemmas. We shall use $C^\dagger$ to indicate the interpretation of a dynamic expression or operator C in *DMG* and $C^\ddagger$ its interpretation in *DMG(2)*. In the first lemma, $\uparrow$ can be instantiated as $\uparrow^\dagger$, $\uparrow^\ddagger$ or $\Uparrow$, and $\downarrow$ as $\downarrow^\dagger$, $\downarrow^\ddagger$ or $\Downarrow$.

Lemma 1

$$\begin{array}{ll} \uparrow x_e = x = \downarrow x_{\uparrow e} & \\ \uparrow {}^\wedge\phi_{\downarrow a} = {}^\wedge\uparrow{}^\vee\,{}^\wedge\phi = {}^\wedge\uparrow\phi & {}^\vee\uparrow\phi_{\downarrow a} = {}^\vee\,{}^\wedge\uparrow{}^\vee\phi = \uparrow{}^\vee\phi \\ \downarrow {}^\wedge\phi_{\uparrow a} = {}^\wedge\downarrow{}^\vee\,{}^\wedge\phi = {}^\wedge\downarrow\phi & {}^\vee\downarrow\phi_{\uparrow a} = {}^\vee\,{}^\wedge\downarrow{}^\vee\phi = \downarrow{}^\vee\phi \end{array}$$

This lemma permits us not to pay attention to objects of type e in the proofs below, and also to pass over $^\wedge$ and $^\vee$, since these operators are completely transparent for the arrows. This leaves only the cases ϕ_t and $\phi_{\langle a,b\rangle}$ to be considered.

Fact 5.1 ($\downarrow\uparrow$-elimination)
$\downarrow\uparrow\phi = \phi$ where $\downarrow\uparrow$ is either $\downarrow^\dagger\uparrow^\dagger$, $\downarrow^\ddagger\uparrow^\ddagger$, or $\Downarrow\Uparrow$

The fact is proved by a simple induction on the type of ϕ. Base cases:

$$\begin{aligned}
\downarrow^{\dagger}\uparrow^{\dagger}\phi_t &= (\lambda q\ \phi \wedge {}^{\vee}q)({}^{\wedge}\mathtt{true}) = \phi \\
\downarrow^{\ddagger}\uparrow^{\ddagger}\phi_t &= (\lambda R\ {}^{\vee}R({}^{\wedge}\phi))({}^{\wedge}\lambda p\ {}^{\vee}p) = \phi \\
\Downarrow\Uparrow\phi_{\tau} &= \lambda q\ (\lambda R\ \phi({}^{\wedge}\ {}^{\vee}R({}^{\wedge}\mathtt{true})))(\lambda p\ {}^{\vee}p \wedge {}^{\vee}q) = \phi
\end{aligned}$$

The induction step, similar in all three cases, looks as follows:

$$\downarrow\uparrow\phi_{\downarrow\langle a,b\rangle} = \lambda x\ \downarrow\uparrow(\phi(\downarrow\uparrow x)) =_{ih} \phi$$

Lemma 2 $\Downarrow\uparrow^{\ddagger}\phi = \uparrow^{\dagger}\phi$; $\downarrow^{\ddagger}\Uparrow\phi = \downarrow^{\dagger}\phi$

Base case:

$$\begin{aligned}
\Downarrow\uparrow^{\ddagger}\phi_t &= \lambda q\ (\lambda R\ {}^{\vee}R({}^{\wedge}\phi))({}^{\wedge}\lambda p\ {}^{\vee}p \wedge {}^{\vee}q) = \lambda p\ \phi \wedge {}^{\vee}p = \uparrow^{\dagger}\phi \\
\downarrow^{\ddagger}\Uparrow\phi_{\tau} &= (\lambda R\ \phi({}^{\wedge}\ {}^{\vee}R({}^{\wedge}\mathtt{true})))({}^{\wedge}\lambda p\ {}^{\vee}p) = \phi({}^{\wedge}\mathtt{true}) = \downarrow^{\dagger}\phi
\end{aligned}$$

Induction:

$$\begin{array}{rlll}
\Downarrow\uparrow^{\ddagger}\phi_{\langle a,b\rangle} = & \lambda x\ \Downarrow\uparrow^{\ddagger}(\phi(\downarrow^{\ddagger}\Uparrow x)) =_{ih} & \lambda x\ \uparrow^{\dagger}(\phi(\downarrow^{\dagger}x)) = & \uparrow^{\dagger}\phi \\
\downarrow^{\ddagger}\Uparrow\phi_{\uparrow^{\dagger}\langle a,b\rangle} = & \lambda x\ \downarrow^{\ddagger}\Uparrow(\phi(\Downarrow\uparrow^{\ddagger}x)) =_{ih} & \lambda x\ \downarrow^{\dagger}(\phi(\uparrow^{\dagger}x)) = & \downarrow^{\dagger}\phi
\end{array}$$

Fact 5.2 $\downarrow^{\ddagger}\phi = \downarrow^{\dagger}\Downarrow\phi$

The fact is proved for all *DMG(2)*-expressions without free variables. However, when we decompose a λ-abstract $\lambda T\ \phi$, we substitute $\uparrow^{\ddagger}T$ for T in the one case, and $\Uparrow\uparrow^{\dagger}$ for T in the other. Therefore, if in the course of induction T is free in Φ, then the induction hypothesis reads

$$\downarrow^{\ddagger}[\uparrow^{\ddagger}T_a/T_{\uparrow^{\ddagger}a}]\Phi \ = \ \downarrow^{\dagger}\Downarrow[\Uparrow\uparrow^{\dagger}T_a/T_{\uparrow^{\ddagger}a}]\Phi$$

The base cases are simple:

$$\begin{gathered}
\downarrow^{\ddagger}[\uparrow^{\ddagger}T_a/T_{\uparrow^{\ddagger}a}]T = T = \downarrow^{\dagger}\Downarrow[\Uparrow\uparrow^{\dagger}T_a/T_{\uparrow^{\ddagger}a}]T \\
\downarrow^{\ddagger}(\uparrow^{\ddagger}\mathsf{con}) =_{6.1} \downarrow^{\dagger}\uparrow^{\dagger}\mathsf{con} =_{2} \downarrow^{\dagger}\Downarrow(\uparrow^{\ddagger}\mathsf{con})
\end{gathered}$$

The fact holds trivially for all dynamic formulas:

$$\downarrow^{\ddagger}\Phi_{\uparrow^{\ddagger}t} = \Phi({}^{\wedge}\lambda p\ {}^{\vee}p) = (\lambda q\Phi({}^{\wedge}\lambda p\ {}^{\vee}p \wedge {}^{\vee}q))({}^{\wedge}\mathtt{true}) = \downarrow^{\dagger}\Downarrow\Phi$$

Induction (we assume ϕ is in normal form, so the head functor in an application is either a constant or a variable):

$$\begin{array}{lrl}
\downarrow^{\ddagger}\lambda T_{\uparrow^{\ddagger}a}\phi = & \lambda T_a\downarrow^{\ddagger}[\uparrow^{\ddagger}T/T]\phi =_{ih} & \\
& \lambda T_a\downarrow^{\dagger}\Downarrow[\Uparrow\uparrow^{\dagger}T/T]\phi = & \downarrow^{\dagger}\Downarrow\lambda T_{\uparrow^{\ddagger}a}\phi \\
 & & \\
\downarrow^{\ddagger}(\uparrow^{\ddagger}\beta(\alpha)) = & \downarrow^{\ddagger}\uparrow^{\ddagger}(\beta(\downarrow^{\ddagger}\alpha)) =_{6.1} & \\
& \downarrow^{\dagger}\uparrow^{\dagger}(\beta(\downarrow^{\ddagger}\alpha)) =_{2} & \\
& \downarrow^{\dagger}\Downarrow\uparrow^{\ddagger}(\beta(\downarrow^{\ddagger}\alpha)) = & \downarrow^{\dagger}\Downarrow(\uparrow^{\ddagger}\beta(\alpha)) \\
 & & \\
\downarrow^{\ddagger}[\uparrow^{\ddagger}T_a/T_{\uparrow^{\ddagger}a}]T_a(\alpha) = & \downarrow^{\ddagger}\uparrow^{\ddagger}(T(\downarrow^{\ddagger}\alpha)) =_{ih} & \\
& \downarrow^{\ddagger}\uparrow^{\ddagger}(T(\downarrow^{\dagger}\Downarrow\alpha)) =_{6.1} & \\
& \downarrow^{\dagger}\Downarrow\Uparrow\uparrow^{\dagger}(T(\downarrow^{\dagger}\Downarrow\alpha)) = & \downarrow^{\dagger}\Downarrow[\Uparrow\uparrow^{\dagger}T_a/T_{\uparrow^{\ddagger}a}]T_a(\alpha)
\end{array}$$

Finally we prove the equivalence of restricted *DMG(2)* and *DMG*:

Fact 5.3 (Translation from *DMG(2)* into *DMG*)
For all *DMG*-expressions Φ: $\Downarrow\Phi^{\ddagger} = \Phi^{\dagger}$

We show that $\Downarrow$ effectively reduces *DMG(2)*-interpretations to *DMG*-interpretations. The fact is proved by induction on the structure of the dynamic expressions. As in the proof of fact 5.2, if T is free in Φ, then the induction hypothesis reads:

$$\Downarrow[\Uparrow T_{\uparrow^\dagger a}/T_{\uparrow^\ddagger a}]\Phi^\ddagger \;=\; \Phi^\dagger$$

Base cases:

$$\Downarrow(\mathbf{con}^\ddagger) = \Downarrow\uparrow^\ddagger\mathbf{con} =_2 \uparrow^\dagger\mathbf{con} = \mathbf{con}^\dagger$$
$$\Downarrow[\Uparrow T_{\uparrow^\dagger a}/T_{\uparrow^\ddagger a}]T_{\uparrow^\ddagger a} = \Downarrow\Uparrow T_{\uparrow^\dagger a} =_1 T_{\uparrow^\dagger a}$$

The induction now passes off smoothly. The translation operator $\Downarrow$ moves into compound *DMG(2)*-expressions, until it confronts an atomic expression, where it collapses, and it leaves *DMG*-expressions and operations behind. Induction:

$$\Downarrow\mathcal{E}d^\ddagger\Phi = \quad \lambda q\ \exists d\Phi(^\wedge\lambda p\ ^\vee p \wedge {}^\vee q) = \quad \lambda q\ \exists d\Downarrow\Phi(q) = \quad \mathcal{E}d^\dagger\Downarrow\Phi$$

$$\Downarrow{\sim}^\ddagger\Phi = \quad \Downarrow\uparrow^\ddagger\neg\downarrow^\ddagger\Phi =_{2,5.2} \uparrow^\dagger\neg\downarrow^\dagger\Downarrow\Phi = \quad {\sim}^\dagger\Downarrow\Phi$$

$$\Downarrow[\Phi\,;^\ddagger\Psi] = \quad \lambda q\ \Phi(^\wedge\lambda p\ ^\vee p \wedge \Psi(^\wedge\lambda p\ ^\vee p \wedge {}^\vee q)) = \quad \lambda q\Downarrow\Phi(^\wedge\Downarrow\Psi(q)) = \quad [\Downarrow\Phi\,;^\dagger\Downarrow\Psi]$$

$$\Downarrow\lambda T_{\uparrow^\ddagger a}\ \Phi = \quad \lambda T_{\uparrow^\dagger a}\Downarrow((\lambda T_{\uparrow^\ddagger a}\ \Phi)(\Uparrow T)) = \quad \lambda T_{\uparrow^\dagger a}\Downarrow[\Uparrow T/T]\Phi$$

$$\Downarrow\uparrow^\ddagger\beta(\alpha) = \quad \Downarrow\uparrow^\ddagger(\beta(\downarrow^\ddagger\alpha)) =_{2,5.2} \uparrow^\dagger(\beta(\downarrow^\dagger\Downarrow\alpha)) = \quad \uparrow^\dagger\beta(\Downarrow\alpha)$$

We turn to the proof of the reduction facts from section 2.3. These facts concern the reduction of the closure of *DMG*-expressions into *DIL*-expressions.

Fact 5.4 $[\Phi\,;\Psi]\,;\Upsilon = \Phi\,;[\Psi\,;\Upsilon]$; $[\mathcal{E}d\Phi]\,;\Psi = \mathcal{E}d[\Phi\,;\Psi]$

Fact 5.5

1. $\downarrow\uparrow\phi = \phi$
2. $\downarrow\lambda T_{\uparrow a}\Phi = \lambda T_a\downarrow[\uparrow T/T]\Phi$; $(\uparrow\phi)(\psi) = \uparrow(\phi(\downarrow\psi))$
3. $\downarrow{}^\wedge\Phi = {}^\wedge\downarrow\Phi$; ${}^\vee\uparrow\phi = \uparrow{}^\vee\phi$
4. $\downarrow{\sim}\Phi = \neg\downarrow\Phi$; $\downarrow\mathcal{E}d\Phi = \exists d\downarrow\Phi$
5. $\downarrow[\uparrow\phi\,;\Psi] = \phi \wedge \downarrow\Psi$

Fact 5.6 ${\sim}\Phi = \uparrow\neg\downarrow\Phi$

Fact 5.7 ${\sim}\Phi\,;\Psi = {\sim}[\Phi\,;\Psi]$

Fact 5.8

1. ${\sim}[\Phi\,;\Psi]\,;\Upsilon = {\sim}\Phi\,;{\sim}[{\sim}\Psi\,;\Upsilon]$
2. ${\sim}[\mathcal{E}d\Phi]\,;\Psi = {\sim}\mathcal{E}d{\sim}[{\sim}\Phi\,;\Psi]$
3. ${\sim}{\sim}\Phi = \Phi$
4. ${\sim}\uparrow\phi = \uparrow\neg\phi$

Fact 5.9 (DMG-reduction) If Φ is a *DMG*-expression in normal form, without free variables, then $\downarrow\Phi$can be reduced to a proper *DIL*-expression by means of 5.4, 5.5 and 5.6
If Φ is an extended *DMG*-expression in normal form, without free variables, then $\downarrow\Phi$can be reduced to a proper *DIL*-expression by means of 5.4, 5.5 and 5.7
If Φ is an *DMG(2)*-expression in normal form, without free variables, then $\downarrow\Phi$can be reduced to a proper *DIL*-expression by means of 5.4, 5.5 and 5.8

The dynamic languages are constructed from (characteristically) lifted *DIL*-constants, (characteristically) typed variables, discourse markers, and application, abstraction, $^\wedge$, $^\vee$, $\sim$, $\mathcal{E}d$ and ;. Since we are dealing with expressions without free variables, and since the reduction of a λ-term $\lambda T\ \Phi$ introduces a substitution $[\uparrow T/T]\Phi$, the induction hypothesis is that if T is free in Φ, then it is $\downarrow[\uparrow T/T]\Phi$ that reduces effectively. If we take a look at the reduction rules, the only expressions that are not dealt with immediately, are:

$$\downarrow(\beta(\alpha));\ \downarrow^\vee\Phi;\ \downarrow[\beta(\alpha)\,;\Psi];\ \downarrow[^\vee\Phi\,;\Psi];\ \downarrow[\mathcal{E}d\Phi\,;\Psi];\ \downarrow[[\Phi\,;\Upsilon]\,;\Psi];\ \downarrow[\sim\Phi;\Psi]$$

In the first case, we know that the head functor is either a constant or a variable, because the expression is in normal form. The reduction then proceeds as follows:

$$\downarrow((\uparrow\mathbf{con}(\alpha_1))\cdots(\alpha_n)) =_{6.4.2} \downarrow\uparrow((\mathbf{con}(\downarrow\alpha_1))\cdots(\downarrow\alpha_n)) =_{6.4.1} ((\mathbf{con}(\downarrow\alpha_1))\cdots(\downarrow\alpha_n))$$
$$\downarrow[\uparrow T_a/T_{\uparrow a}]((T_{\uparrow a}(\alpha_1))\cdots(\alpha_n)) =_{6.4.2} \downarrow\uparrow((T_a(\downarrow\alpha_1))\cdots(\downarrow\alpha_n)) =_{6.4.1} ((T_a(\downarrow\alpha_1))\cdots(\downarrow\alpha_n))$$

So we see that if a lift operates on a functor, it can be pushed out of applications of the functor while distributing lowerings over the arguments of the functor. Of course, an extension operator may occur in the middle of a series of applications, but the reduction rule $^\vee\uparrow\phi = \uparrow^\vee\phi$ guarantees that the lift can move out of it. Only the escape of $\uparrow$ from the intension operator is not ensured by the reduction rules. However, the intension of an expression $^\wedge\beta$ does not apply to argument expresions if its extension $^{\vee\wedge}\beta$ is not taken, and this is excluded by the normal form assumption.

In the second case we have $\downarrow^\vee\Phi$. In this case, the intensional expression Φ can only be an intensional constant or an intensional variable, rather or not applied several times. (The normal form assumption excludes it to be an expression fronted by the intension operator.) As in the case above, we know that the lift of the constant or variable can be moved out of eventual applications and next escape from the intension operator by means of our reduction rules:

$$\begin{aligned}\downarrow^\vee((\uparrow\beta(\alpha_1))\cdots(\alpha_n)) =_{6.4.2}\ &\downarrow^\vee\uparrow((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n)) =_{6.4.3}\\ &\downarrow\uparrow^\vee((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n)) =_{6.4.1}\ ^\vee((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n))\end{aligned}$$

By a similar line of reasoning, we ensure reduction also in the third and fourth case:

$$\downarrow[((\uparrow\beta(\alpha_1))\cdots(\alpha_n))\,;\Psi] =_{6.4.2} \downarrow[\uparrow((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n))\,;\Psi] =_{6.4.5}$$
$$((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n)) \wedge \downarrow\Psi]$$
$$\downarrow[^\vee((\uparrow\beta(\alpha_1))\cdots(\alpha_n))\,;\Psi] =_{6.4.2,6.4.3} \downarrow[\uparrow^\vee((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n))\,;\Psi] =_{6.4.5}$$
$$^\vee((\beta(\downarrow\alpha_1))\cdots(\downarrow\alpha_n)) \wedge \downarrow\Psi]$$

In the next two cases we use the associativity facts:

$$\downarrow[\mathcal{E}d\Phi\,;\Psi] =_{6.3} \downarrow\mathcal{E}d[\Phi\,;\Psi] =_{6.4.4} \exists d\downarrow[\Phi\,;\Psi]$$
$$\downarrow[[\Phi\,;\Upsilon]\,;\Psi] =_{6.3} \downarrow[\Phi\,;[\Upsilon\,;\Psi]]$$

The use of associativity in the last case does not yield immediate reduction, but a finite number of applications of that fact ensures that we end up with a directly reducible formula again.

It is only in the last case that the three systems differ. In *DMG*, the reduction proceeds as follows:

$$\downarrow[\sim\Phi\,;\Psi] =_{6.5} \downarrow[\uparrow\neg\downarrow\Phi\,;\Psi] =_{6.4.5} \neg\downarrow\Phi \wedge \downarrow\Psi$$

In the extended version of DMG we have:

$$\downarrow[\sim\Phi\,;\Psi] =_{6.6} \downarrow\sim[\Phi\,;\Psi] =_{6.4.4} \neg\downarrow[\Phi\,;\Psi]$$

In *DMG(2)*, there are four possibilities (the first case of which, again, covers all cases where the first conjunct is the negation of an atomic formula):

$$\downarrow[\sim\uparrow\phi\,;\Psi] =_{6.7.4} \downarrow[\uparrow\neg\phi\,;\Psi] =_{6.4.5} \neg\phi \wedge \downarrow\Psi$$
$$\downarrow[\sim\sim\Phi\,;\Psi] =_{6.7.3} \downarrow[\Phi\,;\Psi]$$
$$\downarrow[\sim\mathcal{E}d\Phi\,;\Psi] =_{6.7.2} \downarrow\sim\mathcal{E}d\sim[\sim\Phi\,;\Psi] =_{6.4.4} \neg\exists d\neg\downarrow[\sim\Phi\,;\Psi]$$
$$\downarrow[\sim[\Phi\,;\Psi]\,;\Psi] =_{6.7.1} \downarrow\sim[\Phi\,;\sim[\sim\Psi\,;\Psi]] =_{6.4.4} \neg\downarrow[\Phi\,;\sim[\sim\Psi\,;\Psi]]$$

We now give an indication how the following sentences, treated in *DMG* and *DMG(2)*, can be assigned an equivalent interpretation within the framework of *FDMG*. We only name the necessary type changes.

(35) If a farmer owns a donkey, he beats it.

$$[\mathrm{2AR}]([\mathrm{1AR}]([\mathrm{3VR}](\mathit{own}'))); \; [\mathrm{1DAR}](\mathit{if}')$$

(36) Every farmer who owns a donkey beats it.

$$[\mathrm{1AR}]([\mathrm{2VR}](\mathit{own}')); \; [\mathrm{1AR}]([\mathrm{3VR}](\mathit{who}')); \; [\mathrm{1DAR}](\mathit{every}')$$

(37) If a client comes in, you pamper him. You offer him a cup of coffee.

$$[\mathrm{1AR}]([\mathrm{2VR}](\mathit{come\ in}')); \; [\mathrm{2VR}](\mathit{pamper\ him}'); \; [\mathrm{1DAR}]([\mathrm{2GD}_{\hat{}\,T}](\mathit{if}')); \; [\mathrm{1AR}](.\,')$$

(38) Either there is no bathroom here, or it is in a funny place. In any case, it is not on the ground floor.

$$[\mathrm{1VR}_{\langle\hat{}T,t\rangle}]([\mathrm{2VR}](\mathit{here}')); \; [\mathrm{1DGD}_{\hat{}\,T}](\mathit{no\ bathroom}'); \; [\mathrm{2VR}](\mathit{is\ in\ a\ funny\ place}')$$
$$[\mathrm{1AR}]([\mathrm{2GD}_{\hat{}\,T}](\mathit{or}')); \; [\mathrm{1AR}](.\,')$$

(39) If there is a bathroom here, it is in a funny place. In any case, it is not on the ground floor. This example is structurally the same as example 37.

(40) No farmer beats a donkey he owns. He doesn't kick it either.

$$[\mathrm{2VR}_{\langle\hat{}T,t\rangle}]([\mathrm{1VR}_{\langle\hat{}e,\langle\hat{}T,t\rangle\rangle}]([\mathrm{3VR}](\mathit{beat}'))); \; [\mathrm{1GD}_{\hat{}e,\hat{}T}](\mathit{a\ donkey\ he\ owns}')$$
$$[\mathrm{1DGD}_{\hat{}\,T}](\mathit{no\ farmer}'); \; [\mathrm{1AR}](.\,')$$

(41) If a chessbox doesn't contain a spare pawn, then it is taped on top of it.

$$[\mathrm{2VR}_{\langle\hat{}T,t\rangle}]([\mathrm{1VR}_{\langle\hat{}e,\langle\hat{}T,t\rangle\rangle}]([\mathrm{3VR}](\mathit{contain}'))); \; [\mathrm{1GD}_{\hat{}e,\hat{}T}](\mathit{a\ spare\ pawn}')$$
$$[\mathrm{1GD}_{\hat{}\,T}](\mathit{a\ chessbox}'); \; [\mathrm{1DAR}](\mathit{if}'); \; [\mathrm{1DGD}_{\hat{}\langle\langle\hat{}e,\langle\hat{}T,t\rangle\rangle,\langle\hat{}T,t\rangle\rangle,\hat{}T}](\mathit{not}')$$

References

[1] Asher, N. & Wada, H., 1988, 'A computational account of syntactic, semantic and discourse principles for anaphora resolution', *Journal of Semantics* 6, 309–344

[2] Barwise, J., 1987, 'Noun phrases, generalized quantifiers and anaphora', in: Gärdenfors, P. (ed.) *Generalized quantifiers*, Dordrecht: Reidel, 1–29

[3] Barwise, J. & Perry, J., 1983, *Situations and attitudes*, Cambridge, Mass.: Bradford Books

[4] Benthem, J. van 1986, *Essays on logical semantics*, Dordrecht: Reidel

[5] Berg, M. van de, 1990, 'A dynamic predicate logic for plurals', in: Stokhof, M. & Torenvliet, L., *Proceedings of the seventh Amsterdam colloquium, December 19–22, 1989*, Amsterdam: ITLI

[6] Chierchia, G., 1988, 'Dynamic generalized quantifiers and donkey anaphora', in: Krifka, M. (ed.), *Genericity in natural language*, Tübingen: SNS

[7] Chierchia, G., 1990, 'Anaphora and dynamic logic', this volume

[8] Evans, G., 1980, 'Pronouns', *Linguistic Inquiry*, 11, 337–362 [reprinted in: Evans, G., 1985, *Collected papers*, Dordrecht: Foris, 214–248]

[9] Groenendijk, J. & Stokhof, M., 1988, 'Context and information in dynamic semantics', in: Elsendoorn, B. & Bouma, H. (eds.), *Working models of human perception*, New York: Academic Press, 457–488

[10] Groenendijk, J. & Stokhof, M., 1989, 'Dynamic predicate logic', Amsterdam: ITLI, to appear in *Linguistics and Philosophy*

[11] Groenendijk, J. & Stokhof, M., 1990, 'Dynamic Montague grammar', this volume

[12] Heim, I., 1982, *The semantics of definite and indefinite noun phrases*, dissertation, University of Massachusetts, Amherst

[13] Heim, I., 1983, 'File change semantics and the familiarity theory of definiteness', in: Baüerle, R., Schwarze, Ch. & Stechow, A. von (eds.), *Meaning, use and interpretation of language*, Berlin: De Gruyter

[14] Heim, I., 1990, 'E-type pronouns and donkey anaphora', *Linguistics and Philosophy* 13, 137–178

[15] Hendriks, H., 1988, 'Type change in semantics: the scope of quantification and coordination', in: Klein, E. & Benthem, J. van (eds), *Categories, polymorphism and unification*, Edinburgh/Amsterdam: CCS/ITLI

[16] Hendriks, H., 1990, 'Flexible Montague grammar', in: Hendriks, H. & Moortgat, M., *Theory of flexible interpretation: specification and grammar fragment*, EBRA 3175: Deliverable R1.2.A.

[17] Janssen, T., 1986, *Foundations and applications of Montague grammar*, Amsterdam: CWI

[18] Janssen, T., 1990, 'Models for discourse markers', in: Stokhof, M. & Torenvliet, L., *Proceedings of the seventh Amsterdam colloquium, December 19–22, 1989*, Amsterdam: ITLI

[19] Kadmon, N., 1987, *On unique and non-unique reference and asymmetric quantification*, dissertation, University of Massachusetts, Amherst

[20] Kamp, H., 1981, 'A theory of truth and semantic representation', in: Groenendijk, J., Janssen, T. & Stokhof, M. (eds.), *Formal methods in the study of language*, Amsterdam: Mathematical Centre, 277–322 [reprinted in: Groenendijk, J., Janssen, T. & Stokhof, M. (eds.), 1984, *Truth, interpretation and information*, Dordrecht: Foris, 2–41]

[21] Karttunen, L., 1968, *What makes definite noun phrases definite?*, paper P-3871, Santa Monica, California: The RAND Corporation

[22] Lambek, J., 1958, 'The mathematics of sentence structure'. *American Mathematical Monthly*, 65, 154–169 [reprinted in: Buszkowsky, W., Marciszewski, W. & van Benthem, J. (eds.), 1989, *Categorial grammar*, Amsterdam: Benjamins]

[23] Montague, R., 1974, 'Universal grammar', in: Thomason, R. (ed.), *Formal philosophy*, New Haven: Yale University Press

[24] Montague, R., 1974, 'The proper treatment of quantification in ordinary English', in: Thomason, R. (ed.), *Formal philosophy*, New Haven: Yale University Press

[25] Muskens, R., 1989, *Meaning and partiality*, dissertation, University of Amsterdam

[26] Partee, B. & Rooth, M., 1983, 'Generalized conjunction and type ambiguity', in: Bäuerle, R., Schwarze, Ch. & Stechow, A. von (eds.), *Meaning, use and interpretation of language*, Berlin: De Gruyter

[27] Pelletier, F.J. & Schubert, L., 1988, 'Generically speaking', in: Chierchia, G., Partee, B. & Turner, R (eds.), *Properties, types and meaning* Vol. II, Dordrecht: Reidel

[28] Rooth, M., 1987, 'Noun phrase interpretation in Montague grammar, file change semantics, and situation semantics', in: Gärdenfors, P. (ed.) *Generalized quantifiers*, Dordrecht: Reidel, 237–269

[29] Seuren, P., 1985, *Discourse semantics*, Oxford: Blackwell

[30] Zeevat, H., 1989, 'A compositional approach to discourse representation theory', *Linguistics and Philosophy* 12, 95–131

The ITLI Prepublication Series

1990

Logic, Semantics and Philosophy of Language

LP-90-01 Jaap van der Does	A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof	Dynamic Montague Grammar
LP-90-03 Renate Bartsch	Concept Formation and Concept Composition
LP-90-04 Aarne Ranta	Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn	Nominal Tense Logic
LP-90-06 Gennaro Chierchia	The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia	Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks	Flexible Montague Grammar
LP-90-09 Paul Dekker	The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen	Models for Discourse Markers

Mathematical Logic and Foundations

ML-90-01 Harold Schellinx	Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten	A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema	Relational Games
ML-90-04 Maarten de Rijke	Unary Interpretability Logic
ML-90-05 Domenico Zambella	Sequences with Simple Initial Segments

Computation and Complexity Theory

CT-90-01 John Tromp, Peter van Emde Boas	Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette	A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar	Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Leen Torenvliet	Bounded Reductions

Other Prepublications

X-90-01 A.S. Troelstra	Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke	Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev	On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04	Annual Report 1989
X-90-05 Valentin Shehtman	Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy	Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov	The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev	Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov	On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas	An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone	Provable Fixed points in $I\Delta_0+\Omega_1$, revised version