Institute for Language, Logic and Information

A FORMAL DISCOURSE GRAMMAR TACKLING VERB PHRASE ANAPHORA

Hub Prüst
Remko Scha
Martin van den Berg

ITLI Prepublication Series
for Computational Linguistics CL-91-03

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

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

A FORMAL DISCOURSE GRAMMAR TACKLING VERB PHRASE ANAPHORA

Hub Prüst
Remko Scha
Martin van den Berg
Department of Compuational Linguistics, Faculty of Arts
University of Amsterdam
Spuistraat 134, 1012 VB Amsterdam

ITLI Prepublications
for Computational Linguistics

Received October 1991

A Formal Discourse Grammar tackling Verb Phrase Anaphora

by

Hub Prüst, Remko Scha and Martin van den Berg
Department of Computational Linguistics
Faculty of Arts, University of Amsterdam

Abstract

We argue that an adequate treatment of verb phrase anaphora (VPA) must depart in two major respects from the standard approaches. VP anaphors cannot be resolved by simply identifying an antecedent VP for an anaphoric VP. The resolution process must establish a syntactic/semantic parallelism between larger units (sentences or discourse constituent units) that the VPs occur in. Secondly, discourse structure has a significant influence on the possibilities of VPA. This influence must be accounted for.
We propose a treatment which fulfils these desiderata. It builds on a discourse grammar which incorporates a syntactic/semantic matching procedure that is used to recognize parallel structures in discourse. It turns out that this independently motivated procedure yields the resolution of VPA as a side-effect.

Contents

Overview

This paper is composed as follows. The opening discusses the phenomenon of verb phrase anaphora viewed from a discourse perspective. This perspective highlights some hitherto under-exposed aspects of the VPA problem: parallelism effects and the scope of VPA in relation to the structure of a discourse. In section 2, a central aspect of our approach, namely a mathematical notion of structure sharing, is worked out in detail. Section 3 shifts the attention to the idea of discourse parsing on the basis of a unification grammar, and it presents some relevant discourse grammar rules. Next, VPA resolution receives a treatment based on the formal perspective on discourse described in the first part of the paper (section 4). The last two sections present some ideas about future research and the conclusions of this paper.

1 Verb Phrase Anaphora

1.1 A Discourse Perspective on VPA

Although each utterance in a discoure may be assumed to contribute something to the meaning of that discourse as a whole, the latter cannot be regarded as a simple sum of the meanings of the utterances that constitute the discourse. Utterances, and segments that constitute a discourse in general, are usually involved in complicated mutual dependencies. We believe that a proper understanding of these dependencies in discourse is necessary for a more succesful explanation of linguistic phenomena, such as VPA, which have hitherto been studied mainly at the sentence level.

Parallelism effects have considerable impact on the semantic contribution of an utterance to a discourse. Parallelism effects have been studied in relation to coordination, for instance, in [Lang 77] (page47-61) where two main aspects of parallelism in coordination are distinguished. Firstly, in case of ambiguous conjuncts, ambiguity in the coordination of these conjuncts is reduced because coordination enforces the same reading for all conjuncts ('Selektionseffekt'). Secondly, the reading of an unambiguous conjunct is enforced upon other ambiguous or vague conjuncts in the same coordination ('Übertragungseffect'). To illustrate the first aspect, consider example (1).

(1) (a) *John likes visiting relatives*
(b) *(and) Peter likes visiting friends.*

Both sentences can have an 'active' reading (for (a): John likes to visit relatives) or a 'passive' reading (John likes relatives which visit). However, the meaning of the sequence of (a) and (b) is not simply obtained by all combinations of possible readings for these sentences (which would yield four possible interpretations), but they can only have parallel readings: Either both are interpreted 'active', or both are interpreted 'passive'. Parallelism doesn't allow any 'mixed' interpretation. So it seems that as these sentences may have a common reading, this must be shared. The second aspect of parallelism is illustrated by (2), where sentence (b) enforces the 'active' reading upon (a).

(2) (a) *John likes visiting relatives*
(b) *(and) Peter likes visiting market places.*

(Note that this parallelism effect does not depend on the order of the clauses: if we turn them around, the effect remains the same.) Another example of the latter effect is provided by (3), where the expression in (b) can, in coordination with (a), only be interpereted non-metaphorically.

(3) (a) *John kicked the dog*
(b) *(and) Peter kicked the bucket.*

Another observation along the same lines in [Lang 77] is that in German the sentences (4) may either all express a habit (Fred is a smoker, Peter is a drinker and Roberto a nail-biter) or all express an action (Fred is smoking, Peter is drinking and Roberto is biting his nails).

(4)	(a) *John raucht,*	(a) *John smokes,*
	(b) *Peter säuft,*	(b) *Peter drinks,*
	(c) *Roberto kaut seine Nägel*	(c) *Roberto bites his nails,*

Sentences (a), (b) and (c) all become habitual when succeeded by (4)′(d)

(4)′ (d) ... *und Harry ist drogensüchtig.*
(d) ... *and Harry is a drug addict.*

and all express an action when succeeded by (4)″(d)

(4)″ (d) ... *und Harry schreibt einen Brief.*
(d) ... *and Harry writes a letter.*

These parallelism effects provide one of many reasons why the semantics of a discourse can not be regarded as the simple sum of the semantics of its utterances. They may reduce the number of readings, as in (1), so that readings of ambiguous sentences can not be choosen independently. They may also enforce part of the semantics of an utterance upon other utterances: part of the semantic contribution of (4)″ to discourse (4) is that it enforces an 'actual' reading upon (a) through (c). It is important to notice that, in these cases, the semantic contribution of an utterance to the discourse depends on what it has in common with the other utterances in the discourse segment that it belongs to. As an example we may compare the contribution of the sentence *John hates soccer*, to the contexts (5) and (6).

(5) (a) *Harry likes tennis.*
(b) *John hates soccer.*

(6) (a) *John is a blond.*
(b) *He weighs about 215.*
(c) *He's got a nice disposition.*
(d) *John hates soccer.*

In (5), part of its contribution to this small discourse is the fact that it directs the local topic of the discourse to something like 'boys having some emotional attitude towards sports' (this is what sentence (b) shares with (a)). In (6), the contribution of this sentence to the discourse, however, is totally different: In this context it states yet another quality of John (though of a different kind than the first three).

These considerations show that if we want to account for the mechanisms that function in discourse, it is necessary to develop a device which can detect what two adjacent segments in discourse have in common and what not (i.e. in which respect they are parallel). Among other things, this is important in order to determine the semantic contribution of an utterance to the discourse it appears in. In the sequel of this paper, we shall specify this further in terms of a matching process which yields a structured object (the most specific common denominator) that indicates what part of the syntactic/semantic structure an utterance has in common with the discourse segment it is attached to.

The dependence on the context described above holds more rigorously for utterances which exhibit verb phrase anaphora. VP anaphors induce parallelism in a compelling way. One parallelism effect involved in clauses with VPA is illustrated by (7) and (8).

(7) (a) *Mary likes him*
(b) *(and) Suzy likes him.*

(8) (a) *Mary likes him*
(b) *(and) Fred does too.*

In (7)(a) the pronoun *him* refers to some discourse referent not mentioned in this fragment. The preferred interpretation for (7)(b) is an interpretation that parallels (a), i.e. where both pronouns refer to the same entity. However, there is a possiblity to escape this parallelism: when deictically interpreted, the pronouns can in principle have different referents (for instance, when accompanied by different pointing gestures). In (8), however, it is impossible to escape parallelism because sentence (b) contains a verb phrase anaphor (embodied by the auxiliary *does*). Whatever interpretation the pronoun *him* receives in (8)(a), the interpretation of the VP anaphor in (b) will have to incorporate the same pronoun interpretation. In section 1.3.1, we refer to this aspect of parallelism as 'indexical parallelism'. We shall argue that for instances of VPA which involve both parallel quantifier structures and indexical parallelism, it is necessary to establish syntactic/semantic parallelism between the clauses involved in order to account for these clausal parallelism constraints. This can be obtained by the same general matching process indicated above.

Another important observation with regard to VPA is the fact that antecedents of VPA are not restricted to simplex sentences. A discourse is not just a simple sequence of clauses, but consists of segments that bear specific structural relations to each other. These segments we shall call 'discourse constituent units' (DCUs). VPA antecedents may consist of DCUs that are more complex than one simplex sentence, or they may be embedded in complex DCUs. As an example of a VP anaphor with a complex antecedent, consider (9):

(9) (a) *Fred went to the dentist*
(b) *because he needed a check-up.*
(c) *Sara did too.*

In this example, the VP anaphor may refer to the VPs in both (a) and (b). In that case, the rhetorical relation between the clauses (a) and (b), namely rhetorical subordination indicated by *because*, is preserved in the interpretation of the VP anaphor (so that this anaphor refers to a predication like '(x) went to the dentist because (x) needed a check-up'). Even more interesting are cases like (10), in which we encounter a VP anaphor embedded in a complex DCU.

(10) (a) *Fred went to the dentist*
(b) *because he needed a check-up.*
(c) *Sara did too,*
(d) *but she had to have her wisdom-tooth removed.*

Observe that the contrastive *but* does not contrast (c) with (d), but (a) + (b) with (c) + (d). This gives rise to a binary structure in which these pairs are coordinated. This structure has its impact on the interpretation of the VP anaphor in that it cannot refer to (a) + (b) like in example (9), but only to (a) (Cf. section 1.4).

The observations above give rise to the idea that the problem of VPA must be viewed from a general discoure perspective. In fact, it turns out that an adequate account of the structure of discourse and its semantics (taking into account parallelism effects) yields VPA resolution as a side-effect. It also accounts for the fact that the structure of the discourse has significant influence on the interpretation of VPA. Before we go deeper into these matters, we first shortly discuss some previous approaches to VPA.

1.2 Previous Analyses

The problem of verb phrase anaphora, exemplified in (11), is how to obtain a correct interpretation of an underspecified VP, mostly embodied by an auxiliary such as *does* in (11)(b).[1]

(11) (a) *John plays soccer*
(b) *and Fred does too.*

Previous approaches to VPA, whether formulated within a transformational theory ([Sag 77] and [Williams 77]), a Montagovian framework ([Partee 81]) or Discourse Representation Theory ([Eijck 85] and [Klein 87]), usually treat VP anaphora in terms of 'identity of predication'. This central notion leads to formulating some (copying or deletion) mechanism to interpret semantically identical VPs. Cases like (11) are explained by such a mechanism by interpreting the *does* VP as identical to the VP *plays soccer*. The notion of identity of predication, however, is too narrow to cover clausal parallelism constraints and to explain VPA with complex antecedents. Important constraints on VPA can therefore not be formulated in terms of this notion.

In [Sag 77] and [Williams 77], it has been shown convincingly that VPA is partly a syntactic but for the most part a semantic problem. Some strong syntactic constraints are to be respected. Example (12) illustrates the fact that the auxiliaries must match. The influence of syntactic parallelism is illustrated by (13) and (14) (an example from [Partee 81]). In contrast with (13), the possible antecedent of the VP anaphor in (14) is not

[1] Although the role of the word *too* is important, we shall not deal with it in this paper.

a syntactic VP, which blocks the anaphoric reference.

(12) (a) *Bill can do the shopping.*
(b) *Jerry does too.* *

(13) (a) *Almost every student submitted a paper,*
(b) *but Bill didn't.*

(14) (a) *A paper was submitted by almost every student,*
(b) *but Bill didn't.* *

In Sag's inspiring and extensive study, VPA is characterized in a transformational framework in terms of configurations that allow deletion of identical VPs. VP identity is formulated in terms of 'alphabetic variance'. Any VP whose logical form is an alphabetic variant of that of a preceding VP can be deleted. The notion of alphabetic variance is defined as follows: Two formulas ϕ and ψ are alphabetic variants of each other iff ϕ and ψ are identical up to the renaming of bound variables and any variable x which is free in ϕ and ψ is bound by the same external operator. This notion covers important parallelism aspects of VPA. However, in section 1.1 we have already seen that parallelism effects are not limited to the VPs involved. We shall elaborate further on this problem in section 1.3.

[Williams 77] employs an approach to VP anaphora in the transformational framework that involves a VP rule for the interpretation of a phonologically null VP in the syntactic surface structure (instead of deletion). He proposes a grammar that is composed of two distinct subgrammars: a Sentence Grammar and a Discourse Grammar. Discourse rules only apply to the logical forms provided by the Sentence grammar. Williams sees the VP rule as a rule of the Discourse grammar. However, the notion of Discourse Grammar is not worked out at all. Williams states that "What is urgently needed is a careful articulation of the discourse component of the grammar, ..." ([Williams 77], page 138).

Another interpretative approach to ellipsis is presented in [Partee 81]. Partee & Bach build on the work of [Sag 77] and [Williams 77], based on the notion of identity of predication, but place their treatment in a Montagovian framework. They encounter a fundamental problem which leads to the conclusion "that there is no semantic value that can be assigned to VPs such that VP deletion can be characterized in terms of identity". Consider an instance of the problem in (15) ([Partee 81], page 466]).

(15) (a) *Bill believes that Sally will marry him,*
(b) *but everyone knows that she won't.*

The VP anaphor in (b) must be interpreted as *marry him,* with *him* refering to Bill; *him* can not be bound by *everyone.* So the referential pronoun *him* should not be considered as a 'free variable within the VP' but as a context-dependent element which must be resolved (refer to Bill) before the interpretation of the VP anaphor can be obtained. Partee & Bach argue that, for cases such as (15), it is necessary to formulate a restriction which involves global properties of the logical representation, which is undesirable. We hope to show that from a discourse point of view, such a restriction is not as ad hoc as it may seem because it is a correlate of general discourse processes.

A proposal by Sag et al. ([Sag 85]) for the treatment of VPA in a GPSG framework comes close to the general idea of our approach: "The interpretation of an elliptical construction is obtained by uniformly substituting its immediate constituents into some immediately preceding structure, and computing the interpretation of the results." ([Sag 85], page 162). Sag et al. state that " ... the treatment of verbal ellipsis sketched above is highly tentative and incomplete. This is unavoidable, given the rudimentary state of current understanding of the sorts of discourse factors which play a role in these phenomena." Our treatment, however, is part of a general approach to discourse parsing.

In [Dalrymple 91], higher-order unification is proposed as a formal means for the interpretation of elliptical constructions in general and VPA in particular. The ellipsis resolution problem is subdivided into two separate parts: first, a determination of the parallel structure of the antecedent clause and the anaphoric clause, second, formation of the 'implicit relation' that is needed to resolve the anaphoric clause. The paper is mainly about the formation of the missing predication. This is obtained by an abstract formulation of the problem in terms of equations, and higher order unification to solve these equations. Whereas the present paper deals with the problem of VPA from a discourse perspective, exploiting a unification mechanism as a general means to establish local discourse coherence, [Dalrymple 91] concentrates on the unification mechanism.

The theories mentioned so far try to account for VP anaphora at the sentence level so they don't adress the influence of discourse structure on VPA resolution. Both [Klein 87] and [Eijck 85] give an account of ellipsis in Discourse Representation Theory. They treat ellipsis as an anaphoric relation between a predicate DRS (Discourse Representation Structure) and a discourse referent of the predicate type. The notion of discourse structure in standard DRT, however, is too limited to account for the constraints on the resolution of VP anaphora which result from discourse structure as observed in section 1.1. Neither approach tries to enhance the notion of discourse structure in this respect.

1.3 VPA and Parallelism

VPA induces parallelism between the 'anaphoric clause' (the clause which contains a VP anaphor) and its antecedent. The parallelism induced by VPA is very subtle and, among other things, extends to quantifier structures and to the interpretation of pronouns contained in the antecedent. In this section we shall have a closer look at these two interacting aspects of parallelism.

1.3.1 Structural and Indexical Parallelism

'Structural parallelism' concerns the quantifier structures of the clauses involved. Example (16) illustrates the contribution of structural parallelism to the interpretation of the VP anaphor in question. Sentence (16)(a) may be given a so-called 'collective' interpretation (the full sum that is donated is 10 $) or a 'distributive' reading (each of the three boys donated 10 $).

(16) (a) *Three boys donated 10 $*
(b) *(and) two girls did too.*

The interpretation of (16)(b) is partly determined by the interpretation of (16)(a), in particular a collective (respectively distributive) interpretation of (16)(b) is strongly preferred if (16)(a) is assigned a collective (respectively distributive) interpretation. Two other instances of structural parallelism are shown in (17) and (18).

(17) (a) *Every astronomer observed at least two planets*
(b) *(and) every student did too.*

(18) (a) *Every man likes a woman*
(b) *(and) every boy does too.*

Sentence (18)(a) is well-known for its semantic ambiguity: either the universal quantifier has wide scope or the existential quantier does. If sentence (18)(a) is part of a discourse, the context will probably provide reasons to decide between these two readings (and it will probably also 'narrow down' the interpretation of *every, man* and *woman*). However, lacking context, (18)(a) receives two different interpretations. Consider the continuation of (18)(a) by (18)(b). Whatever the interpretation of (18)(a) may be, the interpretation of (18)(b) will correspond with that interpretation of (18)(a): if (18)(a) has the 'wide scope universal' reading (respectively the 'wide scope existential' reading), (18)(b) has too. The two combinations of interpretations of these sentences where the quantifier scopes differ are ruled out.[2]

Another aspect of parallelism is what one might call 'indexical parallelism': If the antecedent of a VP anaphor contains a pronoun, the pronoun *including its interpretation* must be incorporated in the interpretation of the VP anaphor. The sentences in (19) give a clear example of indexical parallelism. The interpreted pronoun of (a) is in the antecedent of the VP anaphor. Interpreting the anaphor just as *likes him* with some free variable for the pronoun, is obviously wrong because it doesn't provide the full VP anaphor interpretation. (In that case the pronoun might even become bound by the universal term.) So the VP anaphor must include the interpretation of the pronoun.

(19) (a) *Mary likes him*
(b) *(and) every man thinks that Suzy does.*

As another example, consider (20). To obtain the correct interpretation of the anaphoric clause (b), the binding structure of (20) needs to be considered. If *him* in (20)(a) is bound by the existential term *a man*, this must also be so in the interpretation of (b). However, if *him* in (a) is bound by the universal term *everyone*, the pronoun in the VP interpretation of (b) is bound parallel, namely by *John*. (If *him* in (a) refers to some other discourse entity, this interpretation is just 'copied' in the interpretation of the VP anaphor.)

(20) (a) *Everyone told a man that Mary likes him*
(b) *(and) John did too.*

Crucial examples of clausal parallelism are found when both aspects of the

---

[2] [Sag 77] formulates a parallelism requirement on quantifier scope involved in VPA, but his alphabetic variance approach allows only one parallel pair, namely the one where both sentences have a wide scope existential reading. (In the wide scope universal reading, the VPs contain free variables which are bound by different operators outside the VP, which blocks deletion.) This seems to give the wrong result because both parallel readings are possible.

parallelism effects observed here are present at the same time. A paradigmatic example is given below. In this example the VP anaphor induces both structural and indexical parallelism i.e. the interpretation of sentence (b) requires strict parallelism with the interpretation of (21)(a) regarding both quantifier structure and pronoun interpretation.

(21) (a) *Everyone told a man that Mary likes him*
(b) *(and) everyone told a boy that Suzy does.*

Without any context to interpret it in, sentence (21)(a) has six different interpretations, which are listed below in two groups. The difference between the two groups is in the quantifier structure of the formulas.

(21)(a) *Everyone told a man that Mary likes him*
a. $\forall$x[HUMAN(x) --> $\exists$y[MAN(y) & TELL(like(x)(MARY)) (y)(x)]]
b. $\forall$x[HUMAN(x) --> $\exists$y[MAN(y) & TELL(like(y)(MARY)) (y)(x)]]
c. $\forall$x[HUMAN(x) --> $\exists$y[MAN(y) & TELL(like($\underline{he}_1$)(MARY))(y)(x)]]

d. $\exists$y[MAN(y) & $\forall$x[HUMAN(x) --> TELL(like(x)(MARY)) (y)(x)]]
e. $\exists$y[MAN(y) & $\forall$x[HUMAN(x) --> TELL(like(y)(MARY)) (y)(x)]]
f. $\exists$y[MAN(y) & $\forall$x[HUMAN(x) --> TELL(like($\underline{he}_1$)(MARY))(y)(x)]]

Each group is subdivided with regard to the interpretation of the pronoun *him*, which can be interpreted in three different ways, respectively related to *everyone*, to *a man* or to some other discourse referent not mentioned in the sentence. (In the latter case the pronoun is represented by a special variable (a unification variable '$\underline{he}_1$', Cf. section 3). The interpretation of (b) strongly depends on the interpretation of (a). A first incomplete description of the semantic content of (21)(b) will be twofold, depending on the order of quantifiers.

(21)(b) *everyone told a boy that Suzy does.*
A. $\forall$x[HUMAN(x) --> $\exists$y[BOY(y) & TELL($\underline{do}_1$(SUZY))(y)(x)]]

B. $\exists$y[BOY(y) & $\forall$x[HUMAN(x) --> TELL($\underline{do}_1$(SUZY))(y)(x)]]

Now it turns out that the interpretation of (21)(b) must be constructed on the basis of its parallelism with (21)(a), no matter which interpretation(s) the first clause has.

First of all, as in the foregoing examples, the VP anaphor induces structural parallelism: the quantifier structure in the interpretations of the two sentences must be the same. The first interpretation for (b) is only possible if (21)(a) has been assigned an interpretation from the first group; interpretation (b)B. is possible only if (21)(a) has received an interpretation from the second group. Note that, in this case, parallel quantifier structures cannot be obtained by means of some notion of copying: both clauses have their own quantifier structure which will have to be 'checked' for the required parallelism. This example shows that notions of copying or deletion of verb phrases are too limited to account for VPA because clausal contexts need to be checked for certain properties.

Secondly it is important to notice that the interpretation of the pronoun *him* in (21)(a) forces a parallel interpretation in (21)(b). For instance, if *him* refers to some discourse entity 'John' (interpretation c. or f.), the pronoun in the resolution of the VP anaphor must refer to the same entity 'John'. Moreover, if in (21)(a) the pronoun *him* is interpreted as being bound by *a man* (interpretation b. or e.), the interpretation of (21)(b) will have to be parallel, binding the pronoun in the interpretation of the VP anaphor to *a boy*. In order to account for this indexical parallelism, the binding structures (particularly which quantifier binds the pronoun) of the clauses involved have to be parallel. This shows that the notion of VP identity is too narrow as a notion to account for VP anaphora and must be replaced by some notion of clausal parallelism.

The instances of VPA that we have encountered in this section have clearly shown that semantic constraints on VPA are not limited to VP identity but must be stated in terms of parallelism between the anaphoric clause and its antecedent. We provide more evidence for this point of view in the next section where we look at parallel elements outside the VP.

1.3.2 Parallel Elements outside the VP

At least as important as the parallelism between the VPs involved in VPA, is the semantic relation between the elements of an anaphoric clause and their contextual parallels in the antecedent. The elements in the anaphoric clause are related to their antecedent(s) by what on might call 'contrastive kinship': they are semantically related and at the same time contrastive. For resolution, the element(s) of an anaphoric clause must, one way or another, be coupled with their 'parallel(s)' in such a way that these elements stand in this relation of contrastive kinship. The simple fact that the subjects of the sentences in (22), (*John, Bill*) are semantically related entities (i.e. stand in this relation of contrastive kinship) contributes substantially to the acceptability of the anaphoric clause. Compare this with (23), where contrastive kinship between the subjects (*John, the coffee*) is not properly realized, which makes this sequence peculiar.

(22) (a) *John is from Brazil*
(b) *and Bill is too.*

(23) (a) *John is from Brazil*
(b) *and the coffee is too.* ?

Mostly, only non-parallel elements in the antecedent contribute to a full interpretation of the anaphoric clause. For instance, in (24) the locative *in the street* has no parallel in the anaphoric clause and contributes to the interpretation of that clause, whereas it is no part of the interpretation of (25)(b), where it does have a parallel element in the anaphoric clause (*in the park*).[3]

(24) (a) *John jogs in the street,*
(b) *(and) Bill does too.*

---

[3] In general, VPA does not require affixal agreement. However, the interaction between features such as tense with adjuncts is subtle and may, for instance, block adverbial contribution of non-parallel elements to an anaphoric clause. In the following example, the adjunct *last year* does not contribute to the interpretation of the anaphoric clause. *John got married last year. Bill never will.* This provides yet another argument for the matching of semantic structures, but we shall not go into these temporal aspects here.

(25) (a) *John jogs in the street,*
(b) *(and) Bill does in the park.*

(26) (a) *John jogs in the street,*
(b) *(and) Bill does in the morning.* *

The sentences (26) provide another example of the fact that the relation of contrastive kinship between the parallel elements must be properly realized for felicitous use of VPA: (25) is more plausible (at the least) because contrastive kinship is better realized in the pairs (*in the street,in the park*) than in (*in the street,in the morning*). The relevance of this relation with regard to Gapping has already been argued for by [Kuno 76]. Its relevance with regard to VPA has been underestimated.

In many cases the parallel elements are located outside the VP (subjects, as in the previous examples, being the most obvious instance). Non-parallel elements, for that matter, may also be located outside the VP. For instance, *on monday* in (27)(a) is, on many analyses, not part of the VP. However, it does contribute to the interpretation of a VP anaphor in (27)(b).

(27) (a) *John goes to the bank on monday.*
(b) *Bill does too.*

Apart from the agent, unique thematic relations in the antecedent-DCU (such as theme - usually the object case of a verb -, and recipient) always seem to be included in the interpretation of a VP anaphor, i.e. these elements are not allowed to have parallels in the anaphoric clause (contrary to Gapping, Cf. examples (28) through (31)).

(28) (a) *John drinks tea in the morning*
(b) *and Fred does coffee.* *

(29) (a) *John drinks tea in the morning*
(b) *and Fred coffee.*

(30) (a) *John sold a client a table*
(b) *(and) Fred did a chair.* *

(31) (a) *John sold a client a table*
(b) *(and) Fred a chair.*

As observed above, locatives and temporal adjuncts may or may not have parallel elements in the anaphoric clause. They are only part of the relevant predication if they are non-parallel (examples (24) and (25)). The co-occurrence of several locatives and/or temporal adjuncts may give rise to several possible interpretations of an anaphoric clause, such as in examples (32) and (33). (Some speakers seem to view example (32)′ as more acceptable than (32). This indicates that speech stress (in capital letters) must also be considered.)

(32) (a) *John jogs in the street in the morning*
(b) *(and) Mary does in the park.*

(32)′ (a) *John jogs IN THE STREET in the morning*
(b) *(and) Mary does IN THE PARK.*

(33) (a) *John jogs in the street in the morning*
(b) *(and) Mary does in the evening.*

It will be clear that contrastive kinship strongly influences the interpretation of the anaphoric clause in these cases. For instance, the interpretation of (32) in which Mary 'jogs in the park in the morning' is strongly preferred over the interpretation where Mary 'jogs in the street in the park'.[4]

Many treatments of VPA use an indication for the deletion site, which marks the place in the surface structure of the utterance where the ellipsis occurs, e.g. ∅ in (24)′. The examples above show that an approach which uses such a zero element, or a variable, to mark the deletion site, is not in general adequate. In general, it is unknown beforehand (i.e. before resolution) whether the VP anaphor under consideration occupies one or more deletion sites (e.g. example (32)″). Much like in the case of Gapping this is unpredictable.

(24)′ (a) *John jogs in the street*
(b) *(and) Bill does ∅ too.*

(32)″ (a) *John jogs in the street in the morning*
(b) *(and) Mary does ∅ in the park ∅.*

Thus the traditional way to use one zero-element in surface structure to indicate 'the' deletion site is not generally adequate because VPA may involve several deletion sites.

To conclude this section on VPA and parallelism we can say that an adequate treatment of VPA requires the whole anaphoric clause (or DCU) to be matched with its antecedent. Complicated, embedded, VP anaphora in an ambiguous quantifier structure induce clausal parallelism concerning quantifier and binding structures. Furthermore, it is necessary to establish parallel and non-parallel elements (and their mode of combination) which may both be located outside the VP to determine the relevant predication on the basis of parallelism. In the next section we look at instances of VPA which have complex DCUs as their antecedents.

1.4 The Scope of VPA

In previous analyses of VPA, a sentence containing a VP anaphor is assumed to be preceded by one simplex sentence (or conjunct) which incorporates the antecedent. The following examples show that this assumption is not generally correct.

[4] It is interesting to notice that in these examples the anaphoric clause may also be expressed using Gapping, as, for instance, in the example below. (This is where VPA and Gapping meet.)
John jogs in the street in the morning (and) Mary in the park.

(34) (a) *Mary went to Suzy's party yesterday.*
(b) *She had a good time.*
(c) *Fred did too.*

(35) (a) *John went to the library.*
(b) *He borrowed a book on computer science.*
(c) *Bill did too.*

Clauses (34)(c) and (35)(c) are both ambiguous in that the VP anaphor may be interpreted in at least two different ways. The anaphor in (35), for instance, may refer to the VP in (35)(b), or to a conjunction of the VPs in respectively (35)(a) and (35)(b). As it stands, the VP anaphor seems not to be able to refer to the VP in (35)(a) only. This third interpretation, however, is called for in a continuation with a contrastive element such as the word *but* in (36)(d). (Without such a contrastive element, an interpretation with intervening material which is not part of the VPA antecedent seems hard to get.)

(36) (a) *John went to the library.*
(b) *He borrowed a book on computer science.*
(c) *Bill did too,*
(d) *but he borrowed two books on French.*

In one interpretation of (35) the VP anaphor refers to a conjunction of VPs that are distributed over sentences (a) and (b). In this example, however, the VP is part of a somewhat more complex DCU, (c) + (d), which is contrasted with (a) + (b). This determines the structure of the discourse and with that the possible interpretation of the VP anaphor: The structure of discourse (36) is a binary structure that consists of (a) + (b) and (c) + (d). This leads to (a) as possible antecedent clause for the anaphoric clause (c) (Cf. section 4.2).

Example (37) seems to indicate that in complex VPA antecedents, the rhetorical relations between the clauses in the antecedent are preserved in the interpretation of the VP anaphor.

(37) (a) *Fred went to the dentist*
(b) *because he needed a check-up.*
(c) *Sara did too.*

Sentence (a) and (b) stand in a relation of rhetorical subordination, explicitly indicated by *because*. As in (35), the VP anaphor in (37) may refer to (37)(b) alone or to (37) (a) + (b). In the latter case, the rhetorical relation between the clauses is preserved in the VPs that constitute the antecedent of the VP anaphor (which is something like '(x) went to the dentist because (x) needed a check-up'). Again, explicit contrast may disambiguate the VP anaphor:

(38) (a) *Fred went to the dentist*
(b) *because he needed a check-up.*
(c) *Sara did too,*
(d) *but she had to have her wisdom-tooth removed.*

Note that syntactic constraints on VPA such as 'agreement' of auxiliaries seem to hold for the VP of the 'head' of the complex DCU that the anaphoric clause is attached to. Compare for instance (39) and (40). In both examples, the VP anaphor is not ambiguous due to the fact that the auxiliaries in the anaphoric clauses only match one antecedent: in (39), the auxiliaries in (c) (*did*) and (b) (*had*) do not match so the anaphor can only refer to the whole subordination ((a) *because* (b)), whereas in (40) only the proposition in (b) can serve as antecedent of the VP anaphor in (c).

(39) (a) *Fred went to the dentist*
(b) *because he had to have his wisdom-tooth removed.*
(c) *Bill did too.*

(40) (a) *Fred went to the dentist*
(b) *because he had to have his wisdom-tooth removed.*
(c) *Bill had too.*

The following examples also illustrate the important influence of discourse structure on the interpretation of VPA. In (41)(d), the VP anaphor may refer to (c) only, or to (a) through (c). In the latter case, the rhetorical structure of the antecedent is part of the interpretation of the anaphor.

(41) (a) *Harry went to the beach.*
(b) *He didn't go for a swim, however.*
(c) *He took a walk along the seashore.*
(d) *Fred did too.*

(42) (a) *John prepared his exam thouroughly.*
    (b) *He decided not to go on holiday.*
    (c) *He spent the whole week studying.*
    (d) *Every day he went to bed early.*
(e) *Bill did too.*

In (42), the VP anaphor may refer to the whole preceding text, or to the elaboration (42)(b) through (42)(d), or maybe just to the VP in the directly preceding sentence (42)(d). The structure of this small discourse ((42)(b) through (42)(d) being an elaboration) obviously limits the VP anaphor in its reference possibilities. The anaphor cannot refer to (42)(b) or (42)(c) only or only to both.

So it turns out that the reference possibilities of VP anaphora are constrained by the structure of discourse. We give an account of these observations in section 4. Whereas most treatments of VPA view this phenomenon in a too local fashion (namely as only involving an anaphoric VP and a non-complex antecedent VP), we have shown that, in general, the scope of VPA ranges outside VPs and outside clauses. This means that the process of resolving a VP anaphor must involve the entire structure of the clause in which the VP anaphor occurs, as well as the entire structure of the possibly complex antecedent. To put it another way: the traditional notion of 'VP identity' needs to be broadened to a notion of 'DCU parallelism'. In order to account for this we propose to match the anaphoric clause with its antecedent DCU.

2 Calculating the Common Denominator

It has become clear that in order to account for parallelism and discourse semantics, we shall need some mechanism to establish what utterances and discourse units have in common and what not. In this section we shall articulate a device which accomplishes this by matching syntactic/semantic structures and calculating their so-called 'most specific common denominator' (MSCD).

In section 2.1, we first define the syntactic/semantic representations of sentences. These structures resemble semantic derivation trees in the Montague tradition.

Subsequently we define the general notion of a most specific common denominator, which is applicable to arbitrary structures. The notion of a most specific common denominator of two structures is intended as a formalization of the intuitive idea of 'what the two structures have in common'. MSCD may be thought of as unifying those atomic terms that unify and generalizing the others. In section 2.2, it is shown that MSCD can be defined along the same lines as the familiar notions of most general unification (MGU) and most specific generalization (MSG) (Cf. [Robinson 65], [Siekmann 89]).

In section 2.3, we present some applications of calculating MSCD over structures as defined in the first section. This application of most specific common denominator may be considered as a characterization of local semantic coherence. As has already been noticed by [Hobbs 79], a discourse is not perceived as coherent just because successive utterances are 'about' the same entities.[5] The more comprehensive notion of coherence that we intend to formalize, is related to ideas of [Polanyi 85]. Intuitively, this notion may be conceived as 'making the appropriate generalizations over the meanings of successive discourse constituent units'. Consider, for instance, examples (43) and (44). The two sentences in (43) cohere because both state instances of the schema 'some boy likes some sport'. This generalization is indicated in (c) (the notation is explained furtheron). Similarly, (44)(c) reflects what the two sentences in (44) have in common.

(43)	(a) *John likes soccer.*	(a)'	(like'(soccer'))(john')
	(b) *Harry likes tennis.*	(b)'	(like'(tennis'))(harry')
		(c)	(like'(<u>SPORT'</u>))(<u>BOY'</u>)
(44)	(a) *John likes soccer.*	(a)'	(like'(soccer'))(john')
	(b) *He hates tennis.*	(b)'	(hate'(tennis'))($\underline{he_1}$)
		(c)	(<u>EMOTIONAL-ATTITUDE'</u>(<u>SPORT'</u>))(john')

2.1 Syntactic/Semantic Structures

The structured semantic objects that are subject to the matching process resemble semantic derivation trees in the Montague tradition (Cf. [Dowty 81]). One essential difference with Montague Grammar is that there the level of semantic representation is dispensable. In our approach, however, the semantic representations are used in an essential way in a matching

[5] Hobbs observes that although in *John took a train from Paris to Istanbul. He likes spinach,* 'he' can refer only to John, this discourse is not coherent. In this case, the cohesion between the predicates must be taken into account (and we shall see that the present approach accomplishes this).

process. So, contrary to Montague Grammar, the (syntactic) structure of the semantic expressions is important. One in-essential difference: Montague Grammar builds on intensional logic, we use an extensional logic here. This may be easily changed; however, the reasons for using an intensional logic are not important in the light of the topic of the current paper.

2.1.1 The Logic

The many-sorted type logic L_u that is used is a 6-tuple:

$$\Lambda = \langle Typ, Con, Sor, Var, Un, Rav \rangle$$

Definition 1 (Syntax of L_u)
Typ, the set of types, is the smallest set such that:

1. $e,t \in$ Typ
2. if $a,b \in$ Typ, then $\langle a,b \rangle \in$ Typ

For every $a \in$ Typ:

Con_a is a (finite) set of (non-logical) constants of type a
Sor_a is a set of partially ordered sorts of type a
(it is partially ordered by a relation $\prec$; $a \in Sor_a$, and for every $\sigma \in Sor_a$, $\sigma \prec a$)
Var_a is a (denumerably infinite) set of logical variables of type a
Un_a is a (denumerably infinite) set of unification variables of type a
Rav_a is a (denumerably infinite) set of restricted anonymous variables of type a
$(Un \cap Var \cap Rav) = \emptyset$, i.e. Un, Var and Rav are disjoint)

For every $\sigma \in Sor_a$ and $i \in \mathbb{N}$ there is a restricted anonymous variable $\underline{\sigma}_i \in Rav_a$. A function s assigns to every element of $Con_a \cup Var_a \cup Rav_a$ a sort $\sigma \in Sor_a$.[6]
For $\underline{u}_i \in Rav_a$, $s(\underline{u}_i) = u$

The set of meaningful expressions of type a, ME_a is defined recursively:

1. Every constant and every variable of type a is in ME_a

2. If $\phi \in ME_{\langle a,b \rangle}$ and $\psi \in ME_a$, then $\phi(\psi) \in ME_b$
   $\phi(\psi)$ is also written as $app(\phi,\psi)$

3. If $\phi \in ME_b$ and $x \in Var_a$, then $(\lambda x.\phi) \in ME_{\langle a,b \rangle}$

4. If $\phi \in ME_t$ and $x \in Var_a$, then $(\forall x(\phi)) \in ME_t$

5. If $\phi \in ME_t$ and $x \in Var_a$, then $(\exists x(\phi)) \in ME_t$

End Definition

---

[6] We leave aside for the moment the computation of sorts of complex expressions, which will be required in a more fully developed system.

L_u has a standard semantics. A model M for L_u is an ordered pair <D,I>, where D is a non-empty set (the object domain) and I is an interpretation function. The interpretation function assigns a denotation to each non-logical constant of L_u (names, predicates etc.) and a denotation to each sort in such a way that if expression α has sort σ (i.e. $s(\alpha) = \sigma$), then $I(\alpha) \in I(\sigma)$. Its range of possible denotations is defined as follows:

$$D_e = D$$
$$D_t = \{0,1\}$$
$$D_{<a,b>} = D_b^{\;Da}$$

The assignment function g takes variables as its argument and gives, for each variable of type a, a value in the domain of that type: (g: $Var_a \cup Un_a \cup Rav_a \rightarrow D_a$).

The semantic rules of L_u define recursively for any expression α, the extension ('semantic value') of α with respect to model M and value assignment g, denoted $[\![\alpha]\!]^{M,g}$.

Definition 2 (Semantics of L_u)

1. If α is a non-logical constant or a sort, then $[\![\alpha]\!]^{M,g} = I(\alpha)$
   If α is a variable, then $[\![\alpha]\!]^{M,g} = g(\alpha)$

2. If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_b$, then $[\![\alpha(\beta)]\!]^{M,g}$ is the result of applying the function $[\![\alpha]\!]^{M,g1}$ to the argument $[\![\beta]\!]^{M,g2}$)

3. If $\alpha \in ME_a$ and $v \in Var_b$, then $[\![\lambda v.\alpha]\!]^{M,g}$ is that function $h \in D_a^{\;Db}$ such that for any object d in the domain D_b, it holds that $h(d) = [\![\alpha]\!]^{M,g[v/d]}$

4. If $\phi \in ME_t$ and $v \in Var_a$, then $[\![\forall v(\phi)]\!]^{M,g} = 1$ if and only if for every $d \in D_a$ it holds that $[\![\phi]\!]^{M,g[v/d]} = 1$

5. If $\phi \in ME_t$ and $v \in Var_a$, then $[\![\exists v(\phi)]\!]^{M,g} = 1$ if and only if for at least one $d \in D_a$ it holds that $[\![\phi]\!]^{M,g[v/d]} = 1$

In clause 2 of the definition, g1 and g2 are assignments that behave exactly as g, except for elements of Rav_τ, where they are defined as follows: $g1(\underline{\sigma}_i) = g(\underline{\sigma}_{2i})$; $g2(\underline{\sigma}_i) = g(\underline{\sigma}_{2i+1})$
In clauses 3. through 5., $g[v/d]$ is that value assignment exactly like g with the possible difference that $g[v/d](v)$ is the object d.
End Definition

The effect of clause 2 in this definition with regard to restricted anonymous variables is, that different occurrences of the same restricted anonymous variable are treated as if they were different variables. In the formulas, we may therefore use multiple occurrences of the same variable without creating links between them. In practice, we can therefore leave out the index i. The function of sorts and restricted anonymous variables in the current theory is elucidated in section 2.2.

2.1.2 Translations

The table below gives a representative sample of translations of lexical elements. We use x,y,z as variable names for variables of type e; P,Q,R of type <e,t>; S,T,U of type <<e,t>,t> and K,L,M as variable names for variables of type <<<e,t>,t>,t>.

Lexical Element	Cat	Translation	Type
john	T	$\lambda P.P(JOHN)$	$<<e,t>,t>$
a	DET	$\lambda P \lambda Q.\exists x(P(x) \ \& \ Q(x))$	$<<e,t>,<<e,t>,t>>$
every	DET	$\lambda P \lambda Q.\forall x(P(x) \text{ --> } Q(x))$	$<<e,t>,<<e,t>,t>>$
woman	CN	$\lambda x.WOMAN(x)$	$<e,t>$
jog	IV	$\lambda S \lambda x.S(JOG(x))$	$<<<e,t>,t>,t>$
like	TV	$\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(y)(x)))$ $\lambda S \lambda T.S(\lambda x.T(\lambda y.LIKE(y)(x)))$	$<<<e,t>,t>,<<<e,t>,t>,t>>$
must	AUX	$\lambda K \lambda S.MUST(K)(S)$	$<<<<e,t>,t>,t>,<<<e,t>,t>,t>>$

The twofold translation of transitive verbs, such as *like*, results from attaching quantifier distribution to the verb; this motivated in section 4.1. We shall sometimes abbreviate these translations by primed versions of the lexical elements (such as john' for $\lambda P.P(JOHN)$).

In the translations of clauses, both logical variables and unification variables are used. Unification variables are used for context-dependent elements like VP anaphors and pronouns. (In the semantic representations they are underlined, the anaphor *does*, for instance, is represented as '$\underline{do}_1$'.) Both logical variables and unification variables may be subscripted with some natural number. In principle, a pronoun receives a twofold translation: it is translated into a unification variable ($\underline{he}, \underline{he}_1, \underline{he}_2, ..$) and into a standard (logical) variable ($x, y, z, x_1, y_1, ...$). The latter, of course, only when it is bound by an operator such as a lambda-operator or a quantifier. This twofold translation reflects the dichotomy of 'bound variable' versus 'referential' pronouns. The overhead due to this twofold translation of pronouns can be reduced substantially on the basis of syntactic constraints on the distribution of bound variable versus referential pronouns (Cf. [Reinhart 83]).

2.1.3 Structures

We assume a compositional approach to semantics where each syntactic rule has a corresponding semantic rule which specifies the translation of the output of the syntactic rule in terms of the translation of the input to it. Thus the translation of a complex expression is determined by the translations of its parts and the semantic rules that correspond to the syntactic rules that are used to form the expression.

Definition 3 (Syntactic/Semantic Structures)
The set SS_τ of syntactic/semantic structures over an arbitrary type τ is defined as follows:

(1) If α'_τ is the translation of a lexical item α, then $[\alpha'_\tau] \in SS_\tau$

(2) For any semantic rule T_R in which the semantic operation R operates on the expressions α' and β' (i.e. translations of the syntactic elements α and β, respectively, in the

corresponding syntactic rule S(α)(β)), the semantic structure is:

$[(R, [[\alpha'_{Type\alpha}],[\beta'_{Type\beta}]]\ ] \in SS_\tau$

End Definition

For instance, 'John' is a lexical item. Its translation is: λP.P(JOHN) of type <<e,t>,t>, so $[\lambda P.P(JOHN)_{<<e,t>,t>}]$ is a (basic) semantic structure.

Corresponding to the following syntactic rules,

(I) $\alpha_T + \beta_{IV}$ --> $[_S\ \alpha_T\ \beta_{IV}\ ]$
(II) $\alpha_{TV} + \beta_T$ --> $[_{IV}\ \alpha_{TV}\ \beta_T\ ]$
(III) $\alpha_{DET} + \beta_{CN}$ --> $[_T\ \alpha_{DET} + \beta_{CN}\ ]$

we have the following translations and semantic structures, respectively (α,β translate into α',β'):

(I) Translation: $T_{app}(\alpha')(\beta')$ $(\alpha \in P_{IV}, \beta \in P_T)$
Structure: $[app, [[\alpha'_{<<<e,t>,t>,t>}],[\beta'_{<<e,t>,t>}]]\ ]$

(II) Translation: $T_{app}(\alpha')(\beta')$ $(\alpha \in P_{TV}, \beta \in P_T)$
Structure: $[app, [[\alpha'_{<<<e,t>,t>,<<<e,t>,t>,t>>}],[\beta'_{<<<e,t>,t>>}]]\ ]$

(III) Translation: $T_{app}(\alpha')(\beta')$ $(\alpha \in P_{T/CN}, \beta \in P_{CN})$
Structure: $[app, [[\alpha'_{<<e,t><<e,t>,t>>}],[\beta'_{<e,t>}]]\ ]$

These translations are all based on the semantic operation of Function Application. Here follow some examples of syntactic/semantic structures.

(45) *John jogs*
[app,
 $[[\lambda S.S(\lambda x.JOG(x))_{<<<e,t>,t>,t>}]$,
 $[\lambda P.P(JOHN)_{<<e,t>,t>}]]]$

(46) *He whistles*
[app,
 $[[\lambda S.S(\lambda x.WHISTLE(x))_{<<<e,t>,t>,t>}]$,
 $[\underline{he}_1\ _{<<e,t>,t>}]]]$

(47) *Every man likes a woman* ∀x(MAN(x)-->∃y(WOMAN(y) & LIKE(y)(x)))
[app,
 [[app,
  $[\lambda S\lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}]$,
  $[\lambda Q.\exists y(WOMAN(y)\ \&\ Q(y)))_{<<e,t>,t>}]]$,
 $[\lambda Q.\forall x(MAN(x) \text{-->} Q(x))_{<<e,t>,t>}]]]$

(48) *Every man likes a woman* $\exists y(WOMAN(y)\ \&\ \forall x(MAN(x) \dashrightarrow LIKE(y)(x)))$

$$
\begin{array}{l}
[app, \\
\quad [[app, \\
\quad\quad [\lambda S \lambda T.S(\lambda x.T(\lambda y.LIKE(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}], \\
\quad\quad [\lambda Q.\exists y(WOMAN(y)\ \&\ Q(y)))_{<<e,t>,t>}]], \\
\quad [\lambda Q.\forall x(MAN(x) \dashrightarrow Q(x))_{<<e,t>,t>}]]]
\end{array}
$$

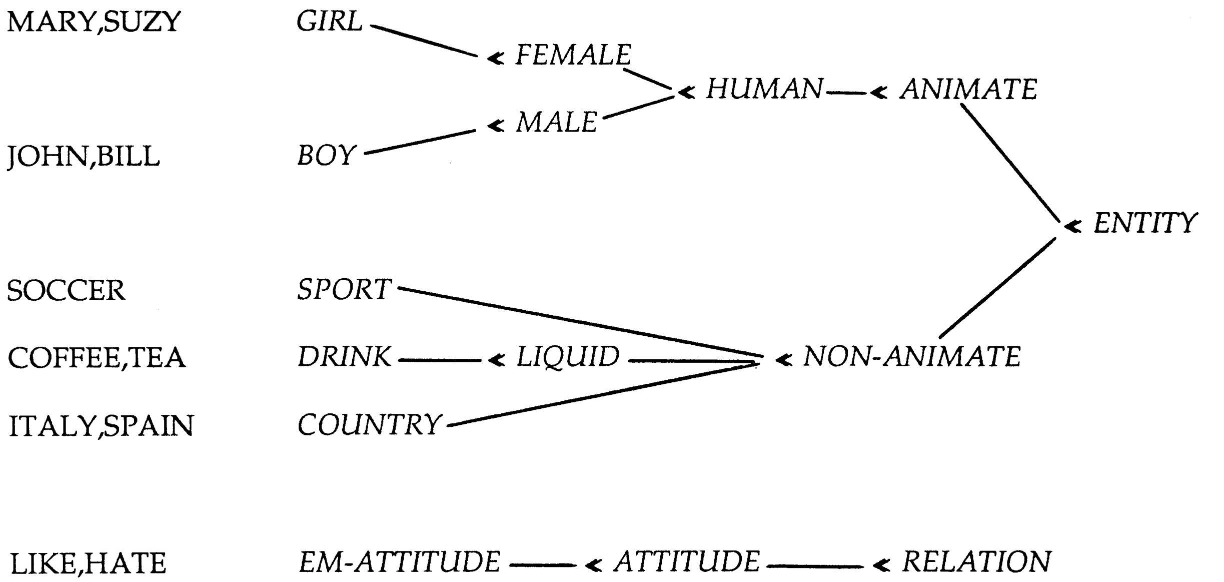
2.2 Most Specific Common Denominator

The definition of MSCD starts with observing that the most specific generalization (MSG) of any unification formalism induces a pre-ordering on its objects:

$$\phi \preceq \psi \text{ iff } MSG(\phi,\psi) = \psi$$

where $\phi \preceq \psi$ is to be read as 'ϕ is at least as specific as ψ'. We choose to go the other way, however, and start with the ordering on the objects and subsequently define the notions of unification and generalization in terms of this ordering.

In the application that we have in mind, a specific consequence of the need to talk about coherence and relatedness of constants is that we want to use a many-sorted logic where relations hold among domains of the (non-disjoint) sorts; this makes it possible to express that something is related to something else in that both belong to the same subset of the domain. It also makes it possible to express that the constant JOHN is more specific than a variable of sort *MALE*, and that a variable of sort *MALE* is more specific than one of sort *HUMAN*, etc. The ordering of sorts is assumed to have a natural, Aristotelean, thesaurus structure, like:[7]



[7] The actual domain of entities and the ordering depend, of course, on the global context of the discourse under consideration.

We thus have, for every type τ, a set of sorts Sor_τ, with a partial order $\prec$ defined on it which has τ itself as its top element. (For instance, if the sort structure depicted above is employed with the type logic defined in section 2.1, we identify the sort *ENTITY* with the type e, and the sort *RELATION* with the type <e,<e,t>>.) The sort function s maps any expression of type τ into an element of Sor_τ.

To exploit the sort structure for articulating MSCDs, we need, for each sort, restricted anonymous variables ranging over it. We thus have defined for every type τ the set Rav_τ: for every sort $\sigma \in Sor_\tau$ and every integer i, we introduce a variable $\underline{\sigma_i} \in Rav_\tau$. We have argued in the previous section that, in practice, we can leave out the index i because multiple occurrences of the same variable are treated independent of each other. Therefore, we simply use underlined sort-names as names for restricted anonymous variables (for instance, the restricted anonymous variable $\underline{\text{BOY}}$ (for $\underline{\text{BOY}_i}$) ranges over the sort *BOY*). (These restricted variables are called 'anonymous' on the analogy with the anonymous variable in PROLOG, because the semantics is defined in such a way that different occurences of the same variable in one expression are independent of each other.)

Before we can give a definition of the ordering on the expressions of the language, some choices with regard to the treatment of variables have to be made.

As for the unification variables, the following problem arises. We want to formulate an inductive definition over the complexity of the formulas, i.e. with axioms of the form

$$\phi(\psi) \preceq \chi(\sigma) \text{ iff } \phi \preceq \psi \text{ and } \chi \preceq \sigma$$

The problem is this: If ϕ and ψ both contain the same unification variable X, how can we make it have the same value on both occasions if they are totally independent occurrences in the recursion? The solution is the standard one: We index the ordering with an assignment ξ that is carried down the induction. We shall write

$$\phi(\psi) \preceq_\xi \chi(\sigma) \text{ iff } \phi \preceq_\xi \psi \text{ and } \chi \preceq_\xi \sigma$$

Such a (unification) assignment ξ is a function from variables of a type τ to meaningful expressions of that same type τ (ξ: $Un_\tau \cup Rav_\tau \to ME_\tau$).[8]

As for the logical variables that also occur in the formulas, there is also a choice to be made. In particular, do we want ∀x.MAN(x) to be equivalent to ∀y.MAN(y)? And if we do, what about free variables, do we want MAN(x) and MAN(y) to be equivalent? The decision is not an easy one, but we think that for formulas of standard logic the former equivalence should hold, but the latter should not. Again our solution follows standard practice. We take an assignment that is to be changed for the actual variable bound in the quantifier clause of the definition. Because we already have defined another kind of assignment, we combine the two in one notion of unification assignment.

---

[8] In the definition of the interpretation of a type logic, an assigment g is carried down the induction in the same way:

$$[\phi(\psi)]^g = [\phi]^g([\psi])^g$$

The difference is that in that case an assignment is a mapping from variables into some domain, whereas the assignment introduced here is a mapping from variables into the language.

Definition 4 (Unification Assignment)
A unification assignment is a function ξ: $Un_\tau \cup Rav_\tau \cup Var_\tau \rightarrow ME_\tau$, such that for any logical variable $x \in Var$, ξ is a morphism into Var: $\xi(x) \in Var$.
End Definition

Now we are ready to define the ordering on the expressions of the type logic L_u defined above.

Definition 5a (Ordering on the language)
Let for any type τ, $a,b \in Con_\tau$, $\underline{u}_i,\underline{v}_i \in Rav_\tau$, $x,y \in Var_\tau$, $X \in Un_\tau$ and $\phi,\psi,\chi,\sigma \in ME_\tau$, the ordering $\phi \preceq_\xi \psi$ (ϕ is a specification of ψ, relative to ξ) of expressions ϕ and ψ of the language be defined as follows:

Atomic Terms
$a \preceq_\xi b$ iff $a \lessdot b$
$\underline{u}_i \preceq_\xi \underline{v}_i$ iff $u \lessdot v$
$a \preceq_\xi \underline{u}_i$ iff $s(a) \lessdot u$
$x \preceq_\xi y$ iff $\xi(x) = \xi(y)$
$\phi \preceq_\xi X$ iff $\xi(X) = \phi$

Logical Expressions
$\neg\phi \preceq_\xi \neg\psi$ iff $\phi \preceq_\xi \psi$
$(\phi \;\&\; \psi) \preceq_\xi (\chi \;\&\; \sigma)$ iff $\phi \preceq_\xi \psi$ and $\chi \preceq_\xi \sigma$
$app(\phi,\psi) \preceq_\xi app(\chi,\sigma)$ iff $\phi \preceq_\xi \psi$ and $\chi \preceq_\xi \sigma$

Quantificational Terms
For these we have to define two auxiliary notions. For two assignments ξ and ζ, we write $\xi \approx_{x,y} \zeta$ if the assignments only differ for the value they assign to x and to y, i.e. for all unification variables X and all logical variables z not equal to x or y: $\xi(X) = \zeta(X)$ and $\xi(z) = \zeta(z)$. For two formulas ϕ and ψ, let $bvar(\phi,\psi)$ denote the set of logical variables that are bound by a quantifier in ϕ or ψ.
For $Q \in \{\forall,\exists,\lambda\}$,

$Qx.\phi \preceq_\xi Qy.\psi$ iff $\exists\zeta \approx_{x,y} \xi$ such that $\zeta(x) = \zeta(y)$ and $\zeta(x) \notin bvar(\phi,\psi)$ and $\phi \preceq_\zeta \psi$

End Definition[9]

---

[9] The fact that we use the definition of the ordering has the advantage that checking the consistency of the induced unification structure amounts to checking whether the resulting structure is a pre-order. This formulation makes it possible to add other properties to the unification without changing any of the objects defined, i.e. it makes extending this formalism beyond term unification as easy as adding extra axioms. Some examples might be:

$\phi \;\&\; \psi \preceq_\xi \psi \;\&\; \phi$	(commutativity)
$(\phi \;\&\; \psi) \;\&\; \sigma \preceq_\xi \phi \;\&\; (\psi \;\&\; \sigma)$	(associativity)
$\phi \;\&\; \psi \preceq_\xi \sigma$ iff $\phi \preceq_\xi \sigma$ and $\psi \preceq_\xi \sigma$	(Left-distributivity)
$\sigma \preceq_\xi \phi \;\&\; \psi$ iff $\sigma \preceq_\xi \phi$ and $\sigma \preceq_\xi \psi$	(Right-distributivity)

In general, adding such axioms will make the computational complexity much higher than the linear complexity of the term unification used in this paper. The point is, however, that we do not need to change any of the notions defined in terms of the order (like MGU, MSG and MSCD).

In terms of this ordering relative to an assignment we can now define the real ordering which will be the basis for the definition of unification and generalization.

Definition 5b (Ordering on the language)
For two arbitrary expressions ϕ and ψ, we write:

$$\phi \preceq \psi \text{ iff there is a } \xi \text{ such that } \phi \preceq_{\xi} \psi$$

$\phi \preceq \psi$ is pronounced: ϕ is at least as specific as ψ
End Definition

Note that the resulting structure is a pre-order because there are expressions which are equivalent under this ordering but not equal. One example is terms which are the same upto unification variable renaming. Another example is two terms with different bound variables. This means that anything defined in terms of this ordering will be unique at most modulo equivalence. This holds in particular for the definition of (most specific) generalization and of (most general) unification. These definitions now amount to writing out the names of the operations:

Definition 6 (MSG)
The most specific generalization (MSG) of ϕ and ψ is an object $\chi := \phi \sqcup \psi$ that is more general than both ϕ and ψ, such that any σ that is more general than both ϕ and ψ is at least as general as χ.
End Definition

Definition 7 (MGU)
The most general unification (MGU) of ϕ and ψ is an object $\chi := \phi \sqcap \psi$ that is more specific than both ϕ and ψ, such that any σ that is more specific than both ϕ and ψ is at least as specific as χ.
End Definition

There is always an MSG, but not always an MGU. If ϕ and ψ do not have a unifier, we sometimes write $\phi \sqcap \psi = \perp$, but this is only notation, $\perp$ is not an element of the pre-order.

Note that the definitions are such that if the same variable occurs in both the left and the right formula, this does not have any special effect. They are not coupled, but regarded as different variables. We think that this is the right behaviour. Variables in unification theory should be used to mark shared locations inside a formula, they should not have some life of their own.

The definition of MSCD is very much like the definitions of MSG and MGU:

Definition 8 (MSCD)
The most specific common denominator (MSCD) $\chi := \phi \not{c}\, \psi$ of ϕ relative to ψ is the most specific generalization of ϕ that still unifies with ψ. In other words, $\chi := \phi \not{c}\, \psi$ is such that it is more general than ϕ, that it unifies with ψ, and that any σ which is also more general than ϕ and unifies with ψ, is at least as general as χ.
End Definition

We sometimes call ϕ in $\phi \not{c}\, \psi$ the default term (and ψ the non-default term).

Digression
This definition is sufficient for our purposes, but it has the disadvantage that the MSCD is not always uniquely defined. If the default term ϕ in $\phi \not{c} \psi$ contains the same unification variable more than twice and the non-default term contains conflicting constants on the same positions, then it might be that resolving this conflict, which is done by taking a somewhat more general term than ϕ, is possible in more that one way, resulting in non-equivalent formulas. For example, say ϕ = a(X,X,X) and ψ = a(b,Y,c), than there are two non-equivalent objects that satisfy the first criterion for being the MSCD (to be more general than ϕ and unify with ψ), namely χ = a(X,X,Z) and χ = a(Z,X,X), but because these are not equivalent, they are each other's counterexamples for the second criterion (that all other terms satisfying the first criterion are either more general or equivalent). A closer analysis of this problem shows that this is a very typical example. The problem arises whenever a variable in the default term connects several places and a conflict is resolved by throwing out one end of that connection. The solution is to realize that what should be thrown out is the whole connection, and not just one end of it. Thus, we rename the structures that (possibly non-uniquely) comply with the definition 8 above as 'weak MSCDs', and formulate a uniquely defined 'strong MSCD' in terms of weak MSCDs.

Definition 9 (Weak and Strong MSCD)
An object χ is said to be a weak common denominator of ϕ and ψ if it is more general than ϕ, unifies with ψ, and no σ that is more specific than χ is also more general than ϕ and unifies with ψ.

The most specific common denominator (or strong common denominator) of ϕ and ψ is an object $\phi \not{c} \psi$ that is the most specific generalization of all weak common denominators of ϕ and ψ.
End Definition

What this does is first look at all objects which are as close to ϕ as is still possible, and then to take the generalization of these in order to throw out accidental connections that still hang around. In the example given above (where ϕ = a(X,X,X) and ψ = a(b,Y,c)), this results in the strong common denominator $\phi \not{c} \psi$ = a(X,Y,Z). This definition seems to be the right one in a great number of cases, including unification of feature structures.[10] Note that when there are no more that two connected variables, which is always the case in the linguistic application described in this paper, strong and weak common denominator are equivalent.
End of Digression

[10] For more details and a comparison of MSCD calculation with Default Unification ([Bouma 90]), we refer to [Berg 90].

2.3 MSCD over Syntactic/Semantic Structures

The units that constitute a discourse may be complex DCUs, contructed from more than one simplex sentence. On the other hand, they may also be formed by sub-sentential objects (for instance in the case of answers to questions, or clauses that exhibit Gapping (which may consist of a sequence of NPs, PPs, etc.)). We restrict ourselves here to examples that consist of pairs of simplex sentences, i.e. the MSCD is calculated on the basis of a matching of clausal structures. Generalization of the matching process as described here to complex structures is straightforward. Generalization to smaller units, however, is all but simple. In section 5, we shall indicate a possible way to define the coupling of structures in the latter case. We shall show that this has in fact much in common with the attachment of DCUs in general.

The most specific common denominator of two syntactic/semantic structures indicates what these two structures have in common. It may also be viewed as a means by which a structure can inherit information such as the (structure of) the antecedents of its underspecified elements (i.e. pronomina and VP anaphora) from earlier structures. The MSCD is calculated over the structure of the incoming utterance and the structure of the DCU it is attached to. As an illustration of the application of common denominator calculation to syntactic/semantic structures, consider examples (49) and (50). (The translation of *like* is somewhat simplified; this has no important consequences because no quantifiers are involved in these examples.)

(49) (a) *John likes Mary*
(b) *(and) Bill likes Suzy.*

[app, S_a
[[app,
$[\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{\langle\langle\langle e,t\rangle,t\rangle,\langle\langle\langle e,t\rangle,t\rangle,t\rangle\rangle}]$
$[\lambda Q.Q(MARY)_{\langle\langle e,t\rangle,t\rangle}]],$
$[\lambda P.P(JOHN)_{\langle\langle e,t\rangle,t\rangle}]]]$

[app, S_b
[[app,
$[\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{\langle\langle\langle e,t\rangle,t\rangle,\langle\langle\langle e,t\rangle,t\rangle,t\rangle\rangle}]$
$[\lambda Q.Q(SUZY)_{\langle\langle e,t\rangle,t\rangle}]],$
$[\lambda P.P(BILL)_{\langle\langle e,t\rangle,t\rangle}]]]$

[app, S_a ¢ S_b (MSCD)
[[app,
$[\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{\langle\langle\langle e,t\rangle,t\rangle,\langle\langle\langle e,t\rangle,t\rangle,t\rangle\rangle}]$
$[\lambda Q.Q(\underline{GIRL})_{\langle\langle e,t\rangle,t\rangle}]],$
$[\lambda P.P(\underline{BOY})_{\langle\langle e,t\rangle,t\rangle}]]]$

The most specific common denominator reflects the fact that JOHN and BILL generalize to a restricted anonymous variable <u>BOY</u> (ranging over the sort *BOY*), and MARY and SUZY generalize to <u>GIRL</u>.

(50) (a) *John likes Mary.*
(b) *Bill adores her.*

[app, S_a
[[app,
$[\lambda S\lambda T.T(\lambda y.S(\lambda x.\text{LIKE}(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}]$
$[\lambda Q.Q(\text{MARY})_{<<e,t>,t>}]]$,
$[\lambda P.P(\text{JOHN})_{<<e,t>,t>}]]]$

[app, S_b
[[app,
$[\lambda S\lambda T.T(\lambda y.S(\lambda x.\text{ADORE}(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}]$
$[\lambda Q.Q(\underline{\text{her}_1})_{<<e,t>,t>}]]$,
$[\lambda P.P(\text{BILL})_{<<e,t>,t>}]]]$

[app, $S_a \not\subset S_b$ (MSCD)
[[app,
$[\lambda S\lambda T.T(\lambda y.S(\lambda x.\underline{\text{EM-ATTITUDE}}(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}]$
$[\lambda Q.Q(\text{MARY})_{<<e,t>,t>}]]$,
$[\lambda P.P(\underline{\text{BOY}})_{<<e,t>,t>}]]]$

As a side-effect of MSCD Calculation, the unification variable $\underline{\text{her}_1}$ unifies with $\lambda P.P(\text{MARY})_{<<e,t>,t>}$.

For convenience, we shall sometimes write $\alpha(\beta)$ as a shorthand for app[α,β], and primed elements as abbreviations for full translations, as illustrated in (51).

(51)	*John likes Mary*	(like'(mary'))(john')	S_a
	Bill adores Suzy	(adore'(suzy'))(bill')	S_b
		($\underline{\text{EMOTIONAL-ATTITUDE}}$'($\underline{\text{GIRL}}$'))($\underline{\text{BOY}}$')	$S_a \not\subset S_b$

3 Discourse Parsing

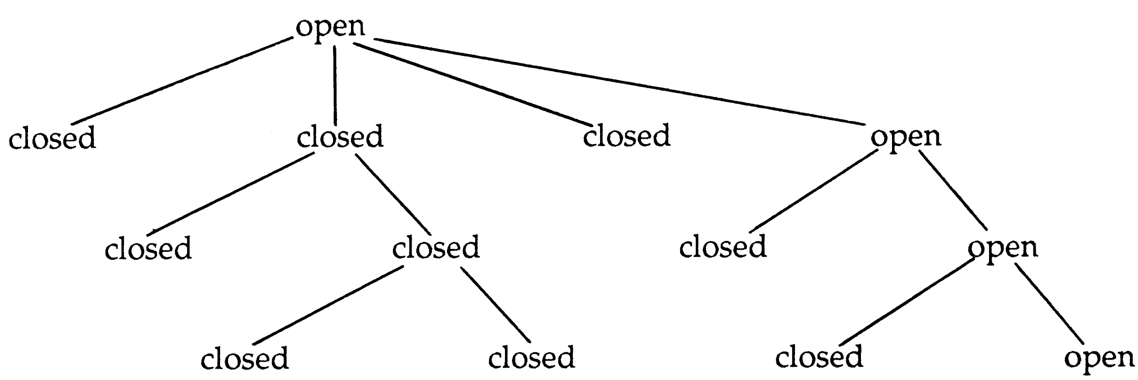
The formalism that we use to describe the hierarchical structure of discourse is a discourse grammar, based on [Scha 88], which consists of a set of explicit rules that assign structural analyses (in the form of tree structures) and meaning representations to discourses. It is a unification grammar: a context free grammar, augmented with attribute/value pairs. The context free rules describe how compound discourse constituent units (DCUs) are built up out of their constituting DCUs.[11]

The grammar comprises rules for building DCUs of different structural types. There are specific rules for different kinds of discourse subordinations (such as, for instance, rhetorical subordinations, elaborations, interruptions) as well as for different kinds of discourse coordinations (for instance, lists, question/answer pairs, narratives). (Cf. [Talmy 78], [Hobbs 79] for definitions of some rhetorical relations and [Mann 82] for a more extensive taxonomy.)

The grammar can be used for discourse parsing, i.e. for assigning structure and semantics to an incoming discourse. Each incoming clause is initially interpreted in a context-independent way. The integration of a clause in its structural context by means of the grammar rules is based on the rhetorical relation that holds between that clause and the DCU it is attached to. Crucial for the incorporation of the new discourse unit in the preceding discourse is the method that we discussed in the previous section for establishing what part of the syntactic/semantic structure it shares with the discourse unit it is attached to.

3.1 General parsing strategy

The discourse parsing process is assumed to be incremental. The incrementality of the parsing process brings with it that at any point in the process, only the right edges in the discourse structure tree are 'open' to form new (sub)structures whereas the rest of the edges are 'closed' as, for instance, in the schematic example below (Cf. [Polanyi 86]). This means that only information on the right edges of the existing discourse tree is addressed.



[11] Nothing hinges on this particular formalism (the CF grammar). In priciple, augmented transition networks, for instance, can do the job as well (Cf [Polanyi 84]). However, the recursive construction of DCUs as constituting units of discourse and also the establishment of local coherence, are crucial to the current approach.

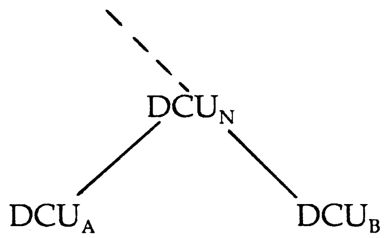
Expansion of the open discourse constituent units is carried out in accordance with the grammar rules. Each rule in the grammar may be used for expanding open edges in a discourse tree. There are essentially two ways to expand an open edge: construct a new DCU by means of a construction rule, or extend an existing DCU by means of an extension rule.

Expansion of an existing edge E by means of a construction rule means: at the place of E, attach the lefthand constituent L of the rule, edge E becomes L's leftmost daughter (the first constituent on the right side of the rule), the rest of the righthand side constituents form the subsequent daughters of E. This is exemplified below.

Old Tree

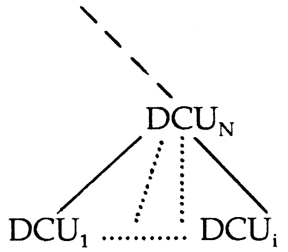
Construction Rule: $DCU_N \rightarrow DCU_A\ DCU_B$
New input instantiates DCU_B

New Tree



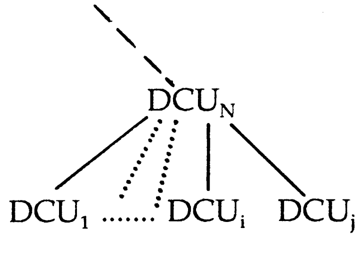
Expanding an edge E by extending an already existing DCU by means of an extension rule comes down to adding a new daughter and updating the semantics of that DCU:

Old Tree



Extension Rule: $DCU_N \Rightarrow DCU_N\ DCU_j$
New input instantiates DCU_j

New Tree



In general, a DCU can only be extended if its schema (i.e. the value of the **schema** attribute which indicates common semantics) is applicable to a new DCU, i.e. if the semantics that is common to all daughters is also present in the new DCU (possibly in an underspecified way). Construction of a DCU, on the other hand, does not necessarily involve instantiation of a schema that is provided by the foregoing discourse. (Some DCUs, however, such as Question-Answer pairs and certain elaborations, require their schema to be instantiated in all daughters; also if new daughters involve construction of new DCUs, Cf. section 3.2)

Each clause is initially interpreted in a context-independent way. This results in the formation of a basic DCU which has as one of its attribute-values a meaning representation which is possibly underspecified for context-dependent elements. The incoming DCU is placed in its (structured) discourse context, thus forming a new (structured) discourse context for a next utterance. Any incoming DCU is placed in the structured context as follows: The incoming DCU is coupled with the 'lowest' DCU that is open for expansion (i.e. the most recently processed open DCU) and that yields a plausible result. The context-dependent interpretation of the incoming DCU is determined by its integration in the preceding (structured) discourse on the basis of the grammar. This is formalized in the grammar rules which describe how the context-depentent interpretation of a clause is formed. At every stage of the parsing process, the semantics of each node in the parse tree is composed of the context-dependent semantics of its daughter nodes.

3.2 Grammar Rules

As mentioned above, the discourse grammar comprises different rules for different structural types. The two main structural types are subordination and coordination. Subordinations may be semantically related (so-called 'semantic subordinations'), or there may be no semantic relation (e.g. interruptions). Semantic subordinations are binary structures in which all or most of the relevant features (reference time, spatial index, modal index etc.) are inherited from the subordinating constituent. Coordinations are n-ary ($n \geq 2$) structures in which all elements have equal status. In this section we discuss a relevant sample of grammar rules. We refer to [Scha 88] for discourse grammar rules (though in a slightly different format) that cover additional rhetorical structures.

In the grammar rules, the following operations are used:

$\alpha \not{c} \beta$ — the most specific common denominator of α and β (this operation is defined in section 2)

$app(\alpha,\beta)$ — Function Application in a typed logic: $\alpha(\beta)$ of type a, if α is of type $<a,b>$ and β of type b,

$\&\&(\alpha,\beta)$ — Generalized Conjunction: $\&\&(\alpha,\beta) = \lambda x_{a1}....\lambda x_{an}.(\alpha(x_{a1})...(x_{an}) \ \& \ \beta(x_{a1})...(x_{an}))$ for α and β of type $<a_1,<a_2,<...,<a_n,t>...>>>$

The first rule in this section is the construction of a so-called 'list' DCU out of other (possibly complex) DCUs. Rule (I) formalizes the idea that a list structure may consist of two DCUs of arbitrary types that have some common semantics. Capital letters in the rule are used to indicate variables (therefore 'DCU1' and 'DCU2' may be of any type). A list DCU comprises several attributes. Only the attributes which are relevant in the context of the current discussion are given in the rule: the **sem** attribute records the context-independent syntactic/semantic representation of the DCU, **consem** records the context-dependent semantics, the **schema** attribute records local semantic coherence. Other attributes like discourse referent sets, reference time, spatial index, modal index etc. are not considered here (but Cf. [Scha 88]).

Rule (I) (List Construction)

list [**schema**:S_1 ¢ S_2, **consem**:&&(S_1,((S_1 ¢ S_2) $\sqcap$ S_2))]
→ DCU1 [**consem**:S_1]
+ DCU2 [**sem**:S_2, **consem**:(S_1 ¢ S_2) $\sqcap$ S_2]

Condition: S_1 ¢ S_2 is a characteristic generalization of S_1 and S_2.

The constraint with regard to the common semantics of a list is implemented as a condition on the applicability of the rule in terms of 'characteristic generalization' i.e. a generalization which incorporates essential characteristics. (This is discussed in section 5). Rule (I) parses sequences such as *'John likes Mary (and) Bill likes Mary'* and *'John likes Mary. Bill does too.'* and *'John likes Mary. Bill adores her.'* and *'John walks. Bill whistles.'* Construction of a list DCU comes about as follows: The context-independent semantics of the second DCU (S_2) is matched with the contextually determined semantics of the first one (S_1). What these structures have in common (i.e. their most specific common denominator, S_1 ¢ S_2) is required to embody a characteristic generalization in order to form a list. The most specific common denominator is stored as the value of the **schema** attribute of the list and it partly determines the syntactic/semantic structure that represents the context-dependent interpretation of DCU2 by unifying the two structures (S_1 ¢ S_2) and S_2. The semantics of the list is taken to be the generalized conjunction of the context-dependent semantics of the constituting DCUs. (N.B. The context-independent semantics of the first DCU and the schema's of DCU1 and DCU2 are omitted because they are irrelevant in this respect. If the first constituting DCU is initial in the discourse, the interpretation of this DCU obviously does not depend on the preceding discourse, therefore the context-dependent semantics (**consem**) of sentences that are initial in a discourse, is taken to be the same as their context-independent semantics (**sem**).)

Consider the following example:

(52) (a) *John likes soccer.*
(b) *Harry hates it.*

As we said above, the context-independent semantics of dcu2, (hate'(<u>it</u>$_1$))(harry'), is matched with the context-dependent semantics of dcu1, (like'(soccer'))(john'), which results in their common denominator S_1 ¢ S_2 : (<u>EMOTIONAL-ATTITUDE</u>'(soccer'))(<u>BOY</u>') which is stored as the value of the **schema** attribute of the list. The following structure represents the corresponding tree:[12]

list1 **schema**: (EMOTIONAL-ATTITUDE'(soccer'))(BOY')
consem: &&((like'(soccer'))(john'),?)
dcu1 **consem**: (like'(soccer'))(john')
dcu2 **sem**: (hate'($\underline{it}_1$))(harry')
consem: ?

The context-dependent interpretation of dcu2 is obtained by unifying the two structures (S_1 ¢ S_2) ⊓ S_2 : (hate'(soccer'))(harry') . The full structure that rule (I) assigns to (52) is thus:

list1 **schema**: (EMOTIONAL-ATTITUDE'(soccer'))(BOY')
consem: (like'(soccer'))(john') & (hate'(soccer'))(harry')
dcu1 **consem**: (like'(soccer'))(john')
dcu2 **sem**: (hate'($\underline{it}_1$))(harry')
consem: (hate'(soccer'))(harry')

Extension rules extend DCUs that have already been built. The following rule extends an n-ary list to an (n+1)-ary (n ≻ 2) list.

Rule (II) (List Extension)

list$_L$ [**schema**:S_1 ¢ S_2, **consem**:&&(S_1,((S_1 ¢ S_2) ⊓ S_2))]
⇒ list$_L$ [**schema**:S, **consem**:S_1]
\+ DCU2 [**sem**:S_2, **consem**:(S_1 ¢ S_2) ⊓ S_2]

Condition: S_1 ¢ S_2 = S

The condition on rule (II) expresses a rigorous requirement on list extension. (One may consider relaxing this restriction to allow more freedom in list extension (for instance by demanding S_1 ¢ S_2 to be a characteristic generalization of the schema S).) As an example of the application of this rule, consider the following discourse:

(53) (a) *John is smoking.*
(b) *Harry is drinking.*
(c) *(and) Bill is writing a letter.*

---

[12] For the sake of clarity, the translations in these examples are somewhat simplified. The syntactic/semantic structures are in fact more fine-grained than is shown here. The values in this example represent the top-level structures of derivation trees (Cf. section 2).

list1 **schema**: (ACTIVITY')(BOY')
consem: (smoke')john' &
(drink')harry' & (write'(letter'))bill'
dcu1 **consem**: (smoke')(john')
dcu2 **sem**: (drink')(harry')
consem: (drink')(harry')
dcu3 **sem**: (write'(letter'))(bill')
consem: (write'(letter'))(bill')

The schema of the list that was constructed for *John is smoking. Harry is drinking.*, namely (ACTIVITY')(BOY'), is applicable to the third utterance *Bill is writing a letter*, so extension of the list is possible according to the rule above. (Tense and aspect features are not discussed here, but it will be clear from the examples in section 1 that lists are constrained in this respect too.)

A specific type of lists is formed by so-called 'topic chains' (Cf. [Polanyi 85]): lists which show a series of predications about the same argument. In the first clause of the chain, the argument is usually expressed by a full NP which occurs in subject position or as a preposed constituent. In the other clauses, it is usually a pronoun in subject position. Rule (III) builds complex predicates (over a sequence of sentences) for chain structures. The reference possibilities of VP anaphora indicate that such complex predicates need to be available for VPA, whereas in other lists, a change of the argument in subjectposition limits the range of VPA. For instance, in *John jogs. He whistles. Bill does too.* The VP anaphor may refer to *whistle* or to *jog and whistle.* (Cf. section 4.2 on complex VPA antecedents. The condition on the applicability of rule (III) expresses a more rigorous requirement than the condition on list rule (I). It formalizes the fact that common denominator calculation over the successive arguments A_1 and A_2 must yield the first argument (A_1).

Rule (III) (Chain Construction)

chain [**schema**:$P_1(A_1)$ ¢ $P_2(A_2)$, **consem**:(&&(P_1,((P_1 ¢ P_2) ⊓ P_2)))(A_1)]
→ DCU1 [**consem**:$P_1(A_1)$]
+ DCU2 [**sem**:$P_2(A_2)$, **consem**:((P_1 ¢ P_2) ⊓ P_2)(A_1)]

Conditions: P_1 ¢ P_2 is a characteristic generalization of P_1 and P_2
A_1 ¢ A_2 = A_1

Rule (III) parses sequences such as *John jogs. He whistles.* and *John likes Mary. He adores her.* For example:

(54) (a) *John jogs.*
(b) *He whistles.*

chain1 **schema**: (ACTIVITY')(john')
consem: (jog' & whistle')(john')

dcu1 **consem**: (jog')(john')
dcu2 **sem**: (whistle')(he_1)
consem: (whistle')(john')

Just like other lists, chaining lists may be extended to n-ary (n > 2) chains. This must be in accordance with rule (IV):

Rule (IV) (Chain Extension)

chain$_L$ [**schema**:$P_1(A_1)$ ¢ $P_2(A_2)$, **consem**:$(\&\&(P_1,((P_1 ¢ P_2) \sqcap P_2)))(A_1)$]
$\Rightarrow$ chain$_L$ [**schema**:$P_0(A_1)$, **consem**:$P_1(A_1)$]
+ DCU2 [**sem**:$P_2(A_2)$, **consem**:$((P_0 ¢ P_2) \sqcap P_2)(A_1)$]

Condition: $P_1(A_1)$ ¢ $P_2(A_2) = P_0(A_1)$

After applying rule (III) to the first two sentences, rule (IV) parse a discourse such as (55) (an example from [Polanyi 85]) into an chain with three constituting DCUs.

(55) (a) *John is a blond.*
(b) *He weighs about 215.*
(c) *He's got a nice disposition.*

For every list expansion, if S_1 and S_2 are the context-independent syntactic/semantic structures of two successive constituting DCUs, the following condition holds:

S_1 must be at least as specific as S_2

We write this as $S_1 \preceq S_2$ (Cf. section 2.2). This condition excludes kataphora in lists as, for instance, in *He walks. John whistles.* and *Bill doesn't. John walks.* However, it does allow *John got mad yesterday. He smashed a window. John never behaved aggressively before.*, but only as a structure which consists of two lists (and not one tripartite list). This is in accordance with results in psycholinguistic research which show that renominalization may give rise to the construction of new DCUs, especially if the referent is not ambiguous such as in the last example (Cf. [Vonk 91]).

The following coordinative rule for contrastive structures parses sequences as exemplified by (56).

(56) (a) *John went to the university,*
(b) *but Bill didn't.*

Rule (V) (Contrast pairs)

contrast [**schema**:S_1 ¢ S_2, **consem**:$R(S_1,((S_1 ¢ S_2) \sqcap S_2))$]
$\rightarrow$ DCU1 [**consem**:S_1]
+ DCU2 [**sem**:$\lambda X.R(X,S_2)$, **consem**:$\lambda X.R(X,((S_1 ¢ S_2) \sqcap S_2))$]

Conditions: $R \in$ {but,however,conversely,on the other hand,.....}
S_1 ¢ S_2 is a characteristic generalization of S_1 and S_2
S_2 somehow 'contrasts' its antecedent S_1

The relation of contrast itself is not pursued in this paper, except for one crucial point: we have seen in 1.4 that, in many cases, relations such as constrast do not simply hold between the clauses that they connect; they often hold between higher level, complex DCUs.

Question-Answer pairs are binary structures. Whereas in list-structures the value of the schema attribute is determined by calculation of the common denominator, in Q-A pairs the structure of a question functions as an interpretation schema, an indisputable explicit common denominator, for the interpretation of the answer. Therefore, the semantics of the question (S_Q) in the following rule, is taken as schema for the resulting structure. The schema functions as before in the interpretation of the compound structure, but it is more compelling.

Rule (VI) (Question-Answer pairs)

qua [**schema**:S_Q, **consem**:S_Q ¢ S_A]
→ DCU1 [**mood**:interrogative, **consem**:S_Q]
+ DCU2 [**sem**:S_A, **schema**:S_Q !, **consem**:S_Q ¢ S_A]

Condition: S_Q ¢ S_A only contains basic terms in its terminal nodes

This rule assigns to (57) the structure below.

(57) (a) *Who sold the painting?*
(b) *Fred did.*

qua1 **schema**: (sell'(painting'))(<u>PERSON</u>')
consem: (sell'(painting'))(fred')

dcu1 **mood**: interrogative
consem: (sell'(painting'))(<u>PERSON</u>')
dcu2 **sem**: ($\underline{do_1}$)(fred')
consem: (sell'(painting'))(fred')

The exclamation mark that marks the schema attribute in DCU2 indicates that the structure requires its schema to be instantiated in expansions of DCU2 (also if new daughters involve construction of DCUs such as in (58)). Example (59) shows that in cooperative conversation, the explicitness of a question as an interpretation schema is not always necessary to achieve the same effect.

(58) A: *Who sold the painting?*
B: *Bill did, because Harry wouldn't.*

(59) A: *Someone must have sold the painting.*
B: *Bill did, because Harry wouldn't.*

Our rules explain the correspondance as well as the distinction between (58) and (59). Whereas in (58) the structure of the question functions as a common denominator for the interpetation of the answer, in (59) the interpretation of the second utterance is obtained by calculating the common denominator over the structures involved.

To conclude this section on grammar rules, we give a rule for a subordinate structure in (VII). This rule assigns structure to sequences such as (60) and (61).

Rule (VII) Rhetorical Subordination

retsub [**index**:K, **schema**:S_1 ¢ S_2, **consem**:R(S_1,[((S_1 ¢ S_2) ⊓ S_2)]$_K$)]
→ DCU1 [**index**:K, **consem**:S_1]
+ DCU2 [**sem**:λX.R(X,S_2), **consem**:λX.R(X,((S_1 ¢ S_2) ⊓ S_2))]

Conditions: R ∈ {because,since,.....}
No condition on S_1 ¢ S_2

(60) (a) *Fred went to the dentist*
(b) *because he needed a check-up.*

(61) (a) *Fred went to the beach*
(b) *because the weather was fine.*

Example (61) illustrates that a common semantics established by structure sharing is not crucial to this relation.

4 A Discourse Analysis of VPA

In this section we discuss how VP anaphors with simple or complex antecedents are resolved. We view anaphor resolution as a side-effect of the process of establishing local coherence (along the lines of [Hobbs 79]) and discourse structure. In parsing discourse with a discourse grammar of the sort we exploit here, VPA resolution is indeed a side-effect of the general discourse parsing process. In this approach there is no need to stipulate an autonomous mechanism for resolution because the resolution mechanism is integrated in the formal account of discourse structure and discourse semantics.

A clause that is underspecified by means of VPA is viewed as a complete discourse constituent unit. It is a complete DCU with an incomplete interpretation (just as most clauses which are added to an unfolding discourse). A DCU is 'complete' if it contains sufficient information to interpret it in the light of the foregoing discourse. Whether this is so will have to result from the parsing process. We have seen that a crucial mechanism in the integration of a new utterance within the previous discourse is a mechanism for establishing the (most specific) common denominator of syntactic/semantic structures of adjacent DCUs, which embodies a formalization of a notion of local discourse coherence. Extraction of the common denominator is not only important for the articulation of local coherence, but also for a correct interpretation of VPA.

4.1 Simple Antecedents

The 'missing material' of an anaphoric clause such as (62)(b) is contained in the MSCD of these structures, which is given below. In the MSCD calculation, the unification variable $\underline{do}_1$ unifies with its antecedent λS.S(λx.JOG(x)), and JOHN and BILL generalize to the restricted (anonymous) variable <u>BOY</u> (ranging over the sort *BOY*).

(62) (a) *John jogs.*
(b) *Bill does too.*

[app,
[[$\lambda S.S(\lambda x.JOG(x))_{<<<e,t>,t>,t>}$],
[$\lambda P.P(JOHN)_{<<e,t>,t>}$]]] $\qquad S_a$

[app,
[[$\underline{do}_{1\ <<<e,t>,t>,t>}$],
[$\lambda P.P(BILL)_{<<e,t>,t>}$]]] $\qquad S_b$

[app,
[[$\lambda S.S(\lambda x.JOG(x))_{<<<e,t>,t>,t>}$],
[$\lambda P.P(\underline{BOY})_{<<e,t>,t>}$]]] $\qquad S_a ¢ S_b$ (MSCD)

The full interpretation of the anaphoric clause is reflected by the structure that results from unifying the relevant MSCD with the structure of the (context-independent) anaphoric clause (S_b). This structure is given below:[13]

[app, $(S_a ⊄ S_b) \sqcap S_b$

[[$\lambda S.S(\lambda x.JOG(x))_{<<<e,t>,t>,t>}$],

[$\lambda P.P(BILL)_{<<e,t>,t>}$]]]

It was argued in section 1.3 that VPA induces structural parallelism. For instance, the interpretations of the sentences in (18) must incorporate parallel quantifier structures.

(18) (a) *Every man likes a woman*
(b) *(and) every boy does too.*

The manner in which we propose to account for structural parallelism is to incorporate quantifier distribution in the translation of the main verb of a clause. This idea goes back to the theory of 'Flexible Montague Grammar' ([Hendriks 88]). FMG deals with scope ambiguities as in (18)(a) by flexible typing of the verb *like* so that the landingplaces of quantifiers are attached to the verb. In FMG, every expression is assigned a basic translation with a semantic type that is as 'low' as possible. General rules of raising and lowering, derive translations with higher types from the basic translations. For instance, sentence (18)(a) needs the following basic translations:

every man	$\lambda P.\forall x[MAN(x) \text{-->} P(x)]$
a woman	$\lambda P.\exists y[WOMAN(y) \ \& \ P(y)]$
like	$\lambda y \lambda x.LIKE(y)(x)$

Two nonequivalent ways of raising the lexical translation of *like* lead to the following derived translations:

(a) $\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(y)(x)))$
(b) $\lambda S \lambda T.S(\lambda x.T(\lambda y.LIKE(y)(x)))$

In the present theory, we incorporate these 'lexical ambiguities' directly in the translations of the verb so that translations (a) and (b) are both stored in the lexicon. (This means that we need not incorporate the whole machinery of FMG, which would complicate the matching process.) So, quantifier distribution can be considered a property of the translation of the verb. Therefore structural parallelism may be viewed as a property of the VPs involved. Combining translation (a) with the basic translations of *every man* and *a woman* results in the 'wide scope universal' translation, translation (b) leads to the 'wide scope existential' reading (S and T are variables of NP-type).

Consider the following structures for (18)(a) and (b) and their MSCD:

[13] Although we don't discuss syntactic constraints on VPA (such as voice agreement), it will be clear that specific syntactic constraints can be incorporated in this matching approach in a straightforward manner.

Every man likes a woman $\forall x(MAN(x)\text{-->}\exists y(WOMAN(y) \;\&\; LIKE(y)(x)))$

$[app,$ $\quad S_a$
$\quad [[app,$
$\qquad [\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}],$
$\qquad [\lambda Q.\exists y(WOMAN(y) \;\&\; Q(y)))_{<<e,t>,t>}]],$
$\quad [\lambda Q.\forall x(MAN(x)\text{-->}Q(x))_{<<e,t>,t>}]]]$

$[app,$ $\quad S_b$
$\quad [[\underline{do}_1\ {}_{<<<e,t>,t>,t>}],$
$\quad [\lambda P \forall x.(BOY(x)\text{-->}P(x))_{<<e,t>,t>>}]]]$

$[app,$ $\quad S_a \not{c}\, S_b$ (MSCD)
$\quad [[app,$
$\qquad [\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}],$
$\qquad [\lambda Q.\exists y(WOMAN(y) \;\&\; Q(y)))_{<<e,t>,t>}]],$
$\quad [\lambda Q.\forall x(\underline{MALE}(x)\text{-->}Q(x))_{<<e,t>,t>}]]]$

The MSCD contains the translation of the verb which incorporates the relevant quantifier distribution. As before, the full interpretation of the anaphoric clause is reflected by the structure that results from unifying the common denominator ($S_a \not{c}\, S_b$) with the structure of the anaphoric clause (S_b):

$[app,$ $\quad (S_a \not{c}\, S_b) \sqcap S_b$
$\quad [[app,$
$\qquad [\lambda S \lambda T.T(\lambda y.S(\lambda x.LIKE(x)(y)))_{<<<e,t>,t>,<<<e,t>,t>,t>>}],$
$\qquad [\lambda Q.\exists y(WOMAN(y) \;\&\; Q(y)))_{<<e,t>,t>}]],$
$\quad [\lambda P.\forall x(BOY(x)\text{-->}P(x))_{<<e,t>,t>}]]]$

This structure implies applying the mode of combination of the missing material (i.e. the non-parallel elements) to the (parallel) elements of the anaphoric clause. As the mode of combination is the same as in the antecedent clause, structural parallelism is accounted for. (The interpretation of (18)(a) and (b) where the existential quantifier has wide scope is obtained in the same manner but of course with a different translation of the verb.)

When combined with a proper name *Mary* as in (63), both translations of the verb *like* result in the same verb phrase interpretation. This also holds for the VP anaphor in the second clause.

(63) (a) *Every man likes Mary*
(b) *(and) every boy does too.*

(64) (a) *John likes a woman*
(b) *(and) every boy does too.*

In (64) the two different interpretations of the VP *like a woman* result in one and the same interpretation of the whole sentence: sentence (64)(a) has only one reading (despite the two different VP interpretations of *like a woman*). The second (anaphoric) sentence, however, has two readings, a wide scope and a narrow scope reading for *a woman*. These are obtained by preserving both VP interpretations of the first clause despite the fact that they give rise to the same sentence interpretation.

Note that our approach builds essentially on the compositionality of the semantics (i.e. on the fact that the meaning of a complex expression is a mathematical function of the meaning of its syntactic parts and their mode of combination (Cf. [Janssen 83]). Usually, compositionality is a preliminary requirement for elegance and credibility of a theory. In the current theory, compositionality is more than a philosophically and practically motivated principle; it is essential for an adequate treatment of VPA because this requires to take into account the mode of combination of VPA-antecedents to deal with parallelism. (This becomes especially clear if we observe structural parallelism induced by VPA.)

The treatment of instances of VPA that involve indexical parallelism follows directly from the definition of the matching process (section 2). Consider again example (21), which we have discussed in section 1.3.1.

(21) (a) *Everyone told a man that Mary likes him*
(b) *(and) everyone told a boy that Suzy does.*

Parallel interpretations in which the pronoun *him* is bound by one of the quantifiers are accounted for because MSCD calculation ensures a parallel binding (otherwise it yields the anonymous variable, which means that there is no relevant common denominator); and as quantifier distribution is attached to the main verb, matching the main verbs of the clauses only yields a relevant common denominator in case of parallel quantifier distribution. If *him* refers to some discourse entity, again only parallel interpretation of the VP anaphor is obtained because the syntactic/semantic structure of an incoming DCU is always matched with the structure that represents the context-dependent interpretation of the antecedent. This means that the interpretation of *him* is 'passed through' to the interpretation of the VP anaphor.

Instances of VPA, such as (32), in which several adjuncts are involved, are treated in the same way as the ones above.

(32) (a) *John jogs in the street in the morning,*
(b) *(and) Mary does in the park.*

However, these cases demand a more sophisticated notion of unification. We come back to this in section 5.

4.2 Complex Antecedents

It was shown in section 1.4 that VP anaphors do not always simply refer to a single preceding VP, i.e. their scope is not limited to single propositions, but they may have complex antecedents. In general, the reference possibilities of VPA are constrained by the structural relations in the discourse. Formulating the restrictions on the reference possibilities of VP anaphora that we observed in terms of the parsing process, it turns out that VP anaphora can only refer to open DCUs. If a VP anaphor does not refer to the immedeately preceding sentence DCU, the parsing algorithm, following the general parsing strategy, recursively tries to attach the anaphoric clause to the previous DCU that is open for expansion. This implies that the parsing process possibly results in an interpretation of the VP anaphor where it has a complex (DCU) antecedent. The discourse structure constraint on VP anaphora resolution thus simply follows from the stipulation of incrementality of the parsing process which results in open versus closed discourse constituent units (Cf. section 3.1).

We have argued that discourse (35) gives rise to two possible structures which entail two different interpretations of the VP anaphor.

(35) (a) *John went to the library.*
(b) *He borrowed a book on computer science.*
(c) *Bill did too.*

The anaphoric clause (c) may have (b) as antecedent or it may have the conjunction of (a) and (b) as antecedent DCU. (Just (a) as antecedent seems only possible with explicit contrast, see furtheron.) According to rule (III) of the discourse grammar, the first two sentences are parsed into a chain structure. The common denominator and semantics of this chain are indicated in the structure below:

chain1 **schema**: (ACTIVITY')(john')
consem: (go_to_lib' & borrow'(cs_book'))(john')

dcu1 **sem**: (go_to_lib')(john')
consem: (go_to_lib')(john')
dcu2 **sem**: (borrow'(cs_book'))($\underline{he}_1$)
consem: (borrow'(cs_book'))(john')

Only the right edges of this tree ('chain1 and 'dcu2') are open for expansion. The next (anaphoric) clause may either be attached to the DCU of sentence (b) (i.e. dcu2) or to the DCU labelled chain1. Two possible interpretations of the VP anaphor in (c) result from the two possible ways to incorporate this clause in the structure above.

A first possiblity is to attach it by expanding the edge labelled dcu2 by applying rule (I). This results in the discourse structure below. The VP anaphor is interpreted here as refering to 'borrow a book on computer science'.

chain1 **schema**: (ACTIVITY')(john')
consem: (go_to_lib' & borrow'(cs_book'))(john')

dcu1 **consem**: (go_to_lib')(john')
list1 **schema**: (borrow'(cs_book'))(BOY')
consem: (borrow'(cs_book'))(john') & (borrow'(cs_book'))(bill'))

dcu2 **consem**: (borrow'(cs_book'))(john')
dcu3 **sem**: ($\underline{do}_1$)(bill')
consem: (borrow'(cs_book'))(bill')

The other possible way to build a discourse structure according to the grammar rules is by expanding the edge labelled chain1 to a list, again by appying rule (I). This has as a side-effect that the VP anaphor is interpreted as 'go to the library and borrow a book'.

list1 **schema**: (go_to_lib' & borrow'(cs_book'))(BOY')
consem: (go_to_lib' & borrow'(cs_book'))(john') &
(go_to_lib' & borrow'(cs_book'))(bill')

chain1 **schema**: (ACTIVITY')(john')
consem: (go_to_lib' & borrow'(cs_book'))(john')

dcu1 **consem**: (go_to_lib')(john')
dcu2 **sem**: (borrow'(cs_book'))(he_1)
consem: (borrow'(cs_book'))(john')
dcu3 **sem**: (do_1)(bill')
consem: (go_to_lib' & borrow'(cs_book'))(bill')

Now consider a continuation such as in (36):

(36) (a) *John went to the library.*
(b) *He borrowed a book on computer science.*
(c) *Bill did too,*
(d) *but he borrowed two books on French.*

In this case, the parallelism between (a)+(b) and (c)+(d) must be taken into account. As before, clause (a) and (b) are parsed into the following chain structure:

chain1 **schema**: (ACTIVITY')(john')
consem: (go_to_lib' & borrow'(cs_book'))(john')

dcu1 **sem**: (go_to_lib')(john')
consem: (go_to_lib')(john')
dcu2 **sem**: (borrow'(cs_book'))(he_1)
consem: (borrow'(cs_book'))(john')

Clauses (c) and (d) give rise to the following constrast pair:

con1 **schema**: ?
consem: do_1(bill') but (borrow'(cs_book'))(he_2)

dcu3 **sem**: (do_1)(bill')
consem: ?
dcu4 **sem**: (borrow'(f_books'))(he_2)
consem: ?

The antecedents of the anaphors are not found because kataphoric VPA is not allowed in this type of structure. The explicitly indicated contrast (*but*) gives rise to a structure in which (a)+(b) and (c)+(d) form contrastive DCUs. The DCUs of (a)+(b) and of (c)+(d) are parsed into the following complex structure:

con2 **schema**: (go_to_lib' & borrow'(book'))(BOY')
consem: (go_to_lib' & borrow'(cs_book'))(john') but
(go_to_lib' but borrow'(f_books'))(bill')

chain1 **schema**: (ACTIVITY')(john')
consem: (go_to_lib' & borrow'(cs_book'))(john')

dcu1 **consem**: (go_to_lib')(john')
dcu2 **sem**: (borrow'(cs_book'))(he$_1$)
consem: (borrow'(cs_book'))(john')
con1 **schema**: (ACTIVITY')(bill')
consem: (go_to_lib' but borrow'(f_books'))(bill')

dcu3 **sem**: (do$_1$)(bill')
consem: (go_to_lib')(bill')
dcu4 **sem**: (borrow'(f_books'))(he$_2$)
consem: (borrow'(f_books'))(bill')

5 Future Research

In this section we discuss some possible refinements of the matching process that we have articulated in this paper.

A possible way to formalize the notion of contrastive kinship, may be to refine the ordering of the domain to cover a notion of 'characteristic generalizability'. Two terms are characteristically generalizable if they are generalizable to an object which exhibits important characteristics of both terms. A pair of terms which are characteristically generalizable is preferred over a pair of terms which are generalizable but not characteristically generalizable. Just as an illustation, consider the following ordering.

MARY,SUZY *GIRL* $<_C$ *FEMALE* $<_C$ *HUMAN* $<_C$ *ANIMATE* $<$ *ENTITY*

JOHN,BILL *BOY* $<_C$ *MALE* $<_C$ *HUMAN* $<_C$ *ANIMATE* $<$ *ENTITY*

COFFEE, TEA *DRINK* $<_C$ *LIQUID* $<$ *NON-ANIMATE* $<$ *ENTITY*

Among other things, this ordering implies that the constants JOHN and MARY, via the sorts *BOY* and *GIRL* that are assigned to them, are characteristically generalizable (indicated by $<_C$) to HUMAN (a restricted variable ranging over the sort *HUMAN*), whereas JOHN and COFFEE are only non-characteristically generalizable (to ENTITY) (compare example (23) in section 1.3.2). The notion of characteristic generalizability is in fact probably a matter of degree and dependent on global context.

Untill this point we have assumed that we are matching structures of full sentences - possibly with overt underspecified elements represented by unification variables - in an unequivocal way. In section 2.3 we have indicated that it may be necessary to generalize the matching process to smaller units than clauses. This may be neccesary for cases of Gapping (like in (65)), but also for some instances of VPA.

(65) (a) *John drinks tea,*
(b) *and Fred coffee.*

These instances of VPA are instances like in (32) and (33). We have argued in section 1.3.2 that contrastive kinship strongly influences the interpretation of the anaphoric clauses in these examples.

(32) (a) *John jogs in the street in the morning*
(b) *(and) Mary does in the park.*

(33) (a) *John jogs in the street in the morning*
(b) *(and) Mary does in the evening.*

As in the case of utterances that exhibit Gapping, these instances of VPA cannot be explained by some 'pre-fabricated' structure with holes/variables in it: neither the number, nor the types nor the places of the gaps (the 'deletion sites') can be specified on forehand, they have to fall out of the interpretation process. The fact that real gaps do not involve any overt variables leads to a somewhat different concept of the matching process that is needed, namely to a notion in which the structure of the antecedent is used as a 'default pattern'. The idea is that new information is not only obtained by straightforward unification (for overt underspecification in the case of *does* and *jogs* in (32)/(33)) but also by some default inheritance (Cf. the role of *in the morning* in the interpretation of (32)(b)). In general, the structure of the antecedent discourse unit may be treated as a pattern in which the elements of the new utterance find their parallels on the basis of contrastive kinship, recency and structural accessibility. The context-dependent interpretation of the new utterance is then obtained by substituting the new elements in place of their parallels in the pattern and applying the same mode of combination as in the antecedent. In that case, the syntactic/semantic representation of a preceding DCU is used as a default pattern for the incoming clause. This entails that there is no stipulation of variables or empty structures on the basis of presumed gaps.

A last possible refinement of our approach that we want to suggest here, concerns instances of VPA for which the unification mechanism as it is formulated now is not sufficient because they show a more complicated clausal parallelism. Consider the following example (an example from Tanya Reinhart, p.c.).

(66) (a) *Lucie tells everyone that she loves him,*
(b) *but no one ever believes that she does.*

This interesting instance of indexical parallelism shows that for an adequate treatment of VPA, it must also be possible to match a two place predicate (*believe*) with a three place predicate that has one argument (the one that has no parallel in the anaphoric clause) filled in (*tell everyone*). Under which conditions this is allowed is as yet unclear. This will need further study. In order to handle this kind of parallelism, we shall need an ordering of the meaningful expressions of the language, and not only of the atomic terms. Such an extension need not be difficult, because it is possible without changing any of the notions defined in terms of the order (like MGU, MSG and MSCD) (Cf. footnote 9 in section 2.2).

6 Conclusion

In this paper, we have presented a formal approach to discourse structure and discourse semantics. This approach culminates in the formulation of a set of grammar rules that are used for discourse parsing and a formal mechanism for the establishment of local discourse coherence.

Furthermore, we have shown that this discourse approach provides a fruitful framework to tackle the phenomenon of underspecification in natural language discourse. We have concentrated on the problem of VPA resolution, which is treated as a side-effect of the establishment of coherence. This treatment takes important parallelism effects into account which do not only relate to VPs but also to their contextual elements. The scope of VPA was shown not to be limited to single propositions. Complex antecedents are constrained by the structure of the preceding discourse, which is explained in terms of the attachment possibilities of clauses to the open discourse constituent units.

References

[Berg 90] Martin van den Berg & Hub Prüst, 'Common Ground and Default Unification' in: *Proceedings of the first meeting of Computer Linguists In the Netherlands (CLIN)*, Utrecht, 1990

[Bouma 90] Gosse Bouma 'Defaults in Unification Grammar' in: *Proceedings of the 28th Annual Meeting of the ACL*, Pittsburg, 1990

[Dalrymple 91] Mary Dalrymple, Stuart M.Shieber & Fernando C.N.Pereira 'Ellipsis and Higher-Order Unification', Palo Alto Research Center, Technical Report Series SSL-91-105, 1991

[Dowty 81] David Dowty, Robert Wall & Stanley Peters 'Introduction to Montague Semantics', Reidel Publishing Company, Dordrecht, Holland, 1981

[Eijck 85] Jan van Eijck 'Aspects of Quantification in Natural Language', Ph.D. thesis, 1985

[Hendriks 88] Herman Hendriks 'Type Change Semantics: the Scope of Quantification and Coordination' in: *Categories, Polymorphism and Unification*, E.Klein and J. van Benthem (eds), University of Edinburgh: Centre for Cognitive Science, and Amsterdam: Institute for Language, Logic and Information, 1988

[Hobbs 79] Jerry R.Hobbs, 'Coherence and Coreference' in: *Cognitive Sience* 3, page 67-90, 1979

[Janssen 83] Theo Janssen 'Foundations and Applications of Montague Grammar', Ph.D. Thesis, Mathematisch Centrum, Amsterdam, 1983

[Klein 87] Ewan Klein 'VP Ellipsis in DR Theory' in: *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, J.Groenendijk & D. de Jongh & M.Stokhof (eds), Groningen-Amsterdam Sudies in Semantics (GRASS), Foris, Dordrecht, 1987

[Kuno 76] Susumu Kuno 'Gapping: A Functional Analysis' in: *Linguistic Inquiry 7*, page 300-318, 1976

[Lang 77] Ewald Lang 'Semantik der Koordinativen Verknüpfung', Studia Grammatica XIV, Akademie-Verlag, Berlin, 1977

[Mann 82] W.Mann, C.Matthiessen & S.Thompson 'Rhetorical Structures Report', Manuscript 1982

[Partee 81] Barbara Partee & Emmon Bach 'Quantification, pronouns and VP anaphora' in: *Formal methods in the study of language*, J.Groenendijk, M.Stokhof & T.Janssen (eds), Mathematisch Centrum, Amsterdam, 1981

[Polanyi 84] Livia Polanyi & Remko Scha 'A Syntactic Approach to Discourse Semantics' in: *Proceedings of COLING*, Stanford, California, 1984

[Polanyi 85] Livia Polanyi 'A Theory of Discourse Structure and Discourse Coherence' in: *Papers from the general session of the Chicago Linguistic Society*, CLS 21, page 306 - 322, 1985

[Polanyi 86] Livia Polanyi 'The Linguistic Discourse Model: Towards a Formal Theory of Discourse Structure', BBN Report No. 6409, 1986

[Reinhart 83] Tanya Reinhart 'Coreference and Bound Anaphora', in: *Linguistics and Philosophy 6*, page 47-88, 1983

[Robinson 65] J.A.Robinson, 'A Machine-Oriented Logic Based on the Resolution Principle', in: *Journal of the ACM 12(1)*, 1965

[Sag 77] Ivan Sag, 'Deletion and Logical Form' Ph.D. Thesis University of Pennsylvania, 1977

[Sag 85] Ivan Sag, Gerald Gazdar, Thomas Wasow & Steven Weisler 'Coordination and How to Distinguish Categories', in: *Natural Language and Linguistic Theory 3*, page 117-172, 1985

[Scha 88] Remko Scha and Livia Polanyi 'An Augmented Contextfree Grammar for Discourse' in: *Proceedings of the 12th International Conference on Computational Linguistics (COLING)*, page 22-27, 1988

[Sidner 84] Candace Sidner 'Focusing in the Comprehension of Definite Anaphora' in: *Computational Models of Discourse*, M.Brady & R.Berwick (eds), The MIT Press, Cambridge, Mass., 1984

[Siekmann 89] Jörg Siekmann 'Unification Theory' in: *Journal of Symbolic Computation 7*, page 207-274, 1989

[Talmy 78] Leonard Talmy 'Relations Between Subordination and Coordination', in: *Universals of Human Language*, J.Greenberg, C.Ferguson & E. Moravcsik (eds), Stanford University Press, 1978

[Vonk 91] Wietske Vonk 'Specificity of referring and the comprehension of text structure', in: *Proceedings of the Workshop on Discourse Coherence*, University of Edinburgh, 1991

[Williams 77] Edwin Williams 'Discourse and Logical Form' in: *Linguistic Inquiry 8(1)*, page 101-139, 1977

The ITLI Prepublication Series

LP-90-13 Zhisheng Huang — Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof — Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke — The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast — Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker — Existential Disclosure, Implicit Arguments in Dynamic Semantics

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
ML-90-06 Jaap van Oosten — Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke — A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx — Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani — Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen — Randomness in Set Theory
ML-90-11 Paul C. Gilmore — The Consistency of an Extended NaDSet

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, Edith Spaan, Leen Torenvliet — Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast — Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas — Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets — Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas — Physiological Modelling using RL
CT-90-09 Roel de Vrijer — Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting

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
X-90-12 Maarten de Rijke — Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev — Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova — Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra — Lectures on Linear Logic

1991

Logic, Semantics and Philosophy of Langauge

LP-91-01 Wiebe van der Hoek, Maarten de Rijke — Generalized Quantifiers and Modal Logic
LP-91-02 Frank Veltman — Defaults in Update Semantics
LP-91-03 Willem Groeneveld — Dynamic Semantics and Circular Propositions
LP-91-04 Makoto Kanazawa — The Lambek Calculus enriched with additional Connectives
LP-91-05 Zhisheng Huang, Peter van Emde Boas — The Schoenmakers Paradox: Its Solution in a Belief Dependence Framework
LP-91-06 Zhisheng Huang, Peter van Emde Boas — Belief Dependence, Revision and Persistence

Mathematical Logic and Foundations

ML-91-01 Yde Venema — Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge — On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella — On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx — Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra — History of Constructivism in the Twentieth Century
ML-91-06 Inge Bethke — Finite Type Structures within Combinatory Algebras
ML-91-07 Yde Venema — Modal Derivation Rules
ML-91-08 Inge Bethke — Going Stable in Graph Models
ML-91-09 V.Yu. Shavrukov — A Note on the Diagonalizable Algebras of PA and ZF
ML-91-10 Maarten de Rijke, Yde Venema — Sahlqvist's Theorem for Boolean Algebras with Operators

Computation and Complexity Theory

CT-91-01 Ming Li, Paul M.B. Vitányi — Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitányi — How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitányi — Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan — Census Techniques on Relativized Space Classes
CT-91-07 Karen L. Kwast — The Incomplete Database
CT-91-08 Kees Doets — Levationis Laus
CT-91-09 Ming Li, Paul M.B. Vitányi — Combinatorial Properties of Finite Sequences with high Kolmogorov Complexity
CT-91-10 John Tromp, Paul Vitányi — A Randomized Algorithm for Two-Process Wait-Free Test-and-Set
CT-91-11 Lane A. Hemachandra, Edith Spaan — Quasi-Injective Reductions

Computational Linguistics

CL-91-01 J.C. Scholtes — Kohonen Feature Maps in Natural Language Processing
CL-91-02 J.C. Scholtes — Neural Nets and their Relevance for Information Retrieval
CL-91-03 Hub Prüst, Remko Scha, Martin van den Berg — A Formal Discourse Grammar tackling Verb Phrase Anaphora

Other Prepublications

X-91-01 Alexander Chagrov, Michael Zakharyaschev — The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev — On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev — Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem — Temporal Logic
X-91-06 — Annual Report 1990
X-91-07 A.S. Troelstra — Lectures on Linear Logic, Errata and Supplement
X-91-08 Giorgie Dzhaparidze — Logic of Tolerance
X-91-09 L.D. Beklemishev — On Bimodal Provability Logics for Π_1-axiomatized Extensions of Arithmetical Theories
X-91-10 Michiel van Lambalgen — Independence, Randomness and the Axiom of Choice
X-91-11 Michael Zakharyaschev — Canonical Formulas for K4. Part I: Basic Results
X-91-12 Herman Hendriks — Flexibele Categoriale Syntaxis en Semantiek: de proefschriften van Frans Zwarts en Michael Moortgat
X-91-13 Max I. Kanovich — The Multiplicative Fragment of Linear Logic is NP-Complete
X-91-14 Max I. Kanovich — The Horn Fragment of Linear Logic is NP-Complete
X-91-15 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic, revised version
X-91-16 V.G. Kanovei — Undecidable Hypotheses in Edward Nelson's Internal Set Theory