Agreement Morphology, Argument Structure and Syntax (3. Revision)

Marcus Kracht
Department of Linguistics, UCLA
3125 Campbell Hall
PO Box 951543
Los Angeles, CA 90095-1543
kracht@humnet.ucla.edu

October 3, 2005

Avant Propos

This book grew out of lecture notes for the 1999 ESSLLI summerschool in Utrecht. The core of the material remains more or less as is while other parts have been completely rewritten in light of the facts that I have since then uncovered.

No work starts from zero. In this case, I have benefitted largely from working with Albert Visser and Kees Vermeulen while I was working in a project on the parallels between natural languages and programming languages. Albert Visser's ideas concerning semantics in general and how to set up a really clean framework for dynamic semantics in particular have had a profound impact on me. It has always been his intention to provide a mathematically elegant and sound semantical framework for natural language. Yet, it is one thing to believe that such a framework is possible and another to actually provide it. This book is about how his ideas on semantics can be made fruitful in linguistic theory. I had to sacrifice some features of the original system. My only excuse here is that language just isn't the way we would like it to be. There are many facts to deal with, and they tend to mess up the system a fair bit. There is however also a fair chance that I haven't managed to make things as simple as I could have done and I apologise for that.

This work has been presented on various occasions and in various stages of incarnation in Paris, Tübingen, Potsdam, Berlin, Saarbrücken and Los Angeles. I wish to thank those in the audience who have helped me to bring out my ideas more clearly and who have pointed out numerous deficiencies of earlier versions. My greatest debt in this respect is to Udo Klein who has read the manuscript many times and provided me with long lists of mistakes. His enthusiasm has kept the project going. Further thanks go to Katherine Demuth, Alan Dench, Jan van Eijck, Hans-Martin Gärtner, Willi Geuder, Hubert Haider, Ed Keenan, Ben Keil, Hap Kolb, Anoop Mahajan, Gereon Müller, David Perlmutter, Ed Stabler, Markus

Steinbach, Kees Vermeulen, Albert Visser and Ralf Vogel. I owe special thanks to Markus and Ralf for long discussions on argument structure and polyvalency. I am deeply indebted to Albert and Hans-Martin for their enthusiasm, without which such an endeavour is not possible. Above all, thanks to Johanna Domokos for her patience with me, for helping me with Finnish and Hungarian and her rich knowledge of languages about which I had never heard before.

For the errors that remain I claim responsibility. I appreciate any remarks from my readers, as they will help me to improve on this subject.

Los Angeles, September 2005, *Marcus Kracht*

Introduction

The last 30 or so years have seen an enormous unfolding of formal semantics sparked off by Montague Grammar. Likewise, generative grammar for somewhat longer has been the major syntactic theory. Both have established themselves as something of a lingua franca in linguistics. Yet, there is a problem that besets both of them: they disregard agreement morphology. For both theories, structure is all that counts. A sentence is analysed in structural terms and morphology often appears to be a mere luxury. Both Montague Grammar and generative grammar thus share a disregard for the surface. This is in sharp distinction to the earlier stages of linguistics where form was the primary target of study. Time has come for a synthesis.

The central claim of this book is that there is an interface between syntax and semantics called *argument structure*, whose main responsibility is to declare how its semantic arguments show up on the surface. The argument structure consists of a list of declarations of the following form:

$$\langle x : \triangledown\oslash : \begin{bmatrix} \text{CASE} : & nom \\ \text{NUM} : & pl \end{bmatrix} \rangle$$

Here, x is a variable, $\vartriangle$ and $\oslash$ are declarations about the way the variable is manipulated. $\triangledown$ says that the variable has to come from the complement when merged; $\oslash$ says that the complement is to be found on the left. The remaining part, an attribute value structure, are expectations on the form of the argument. They are phrased as certain attribute value matrices, but may simply be understood as abstract properties. In terms of categorial grammar we are working with a very flat structure; a head can only declare what kinds of arguments it needs and what structural properties they have. There is no recursion: the space of properties of the arguments is finite. Syntax is then reduced to a question of directionality, while morphology plays into the attribute value structure.

This theory assumes no syntactic structure and no movement. However, more realistically one should think of it as a lexicalist theory on a par with categorial grammar. The combinatorics of the words are encoded in the argument structure, and there is nothing beyond it that matters. Having said that, I do believe that such a theory will almost certainly meet great problems. Verb second and other phenomena seem to be beyond the reach of an approach that bases itself purely on concatenation. But there is no reason that it should. Instead, I will propose to liken syntax to morphology in declaring further possibilities to combine on the surface. In particular, if we allow constituents to be discontinuous we can make up for the loss in expressive power of the original system.

Another important issue is computational complexity. Montague Grammar relies on the typed λ-calculus to do the argument handling and variable substitution. It is known that reduction of typed expressions is very expensive unless they are of the form that we assume here: a function being applied to several arguments, none of which is complex. Since Montague Grammar is quite inflexible in the way it handles its arguments a lot of argument shuffling is needed to assume correct processing. This constantly requires applying a function to a dummy variable and reabstracting it. The present framework deliberately makes variable handling more flexible and thereby achieves a flat type structure. The gain is an algorithm that processes sentences in polynomial time, the exponent being quite low.

Below is an summary of the contents of the chapters.

Chapter 1 deals with the basics of Montague Grammar and how the composition of meaning is achieved in it. We shall briefly comment on the problematic aspects of it and look at some refinements, like Combinatory Categorial Grammar. The semantics that we are proposing in the book will not actually use the recursive type structure. There will be no way to form new types as there is in categorial grammar. However, there will be plenty of basic types (alias sorts) to arrange for the fact that there are objects of different nature. It is to be expected that the basic ontology will consist of types that are considered higher order (sets and properties), but there will be no constructs to create types ad infinitum. We shall therefore talk of sorts rather than types.

Chapter 2 introduces a new kind of semantic representation, based on referent systems, as introduced by Kees Vermeulen. Referent systems treat variables as anonymous; during the merge of two semantical representations, the names that they have in each representation cannot be shown to the outside. There is however an agreed set of so-called *names*, by which variables can be identified under merge. We shall assume that the names are principally form related; that is to say, they contain information about the morphological shape of the sign. Additional information is the sort of the variable and the direction where the sign is found. For example, the variable of the subject of a sentence in German is the one carrying nominative case, while in English it is both case (for pronouns) and the fact that it is to the left of the verb. The collection of statements that tell us which variable is identified under what name is called *argument structure*.

Chapter 3 deals with the question of how it is that morphology can shape the name of variables in a representation. We shall assume that lexical roots contain only a minimum of information on names. Most names are added in the process of forming the actual word. This shall give flexibility in the names under which various words expect their variables. A case in point is diathesis; by applying diathesis to a verbal root before the actual case requirements are being fixed we can account for the different case marking pattern in passives. Furthermore, it follows that agreement morphology overtly expresses the form requirements of the head for its arguments. In addition, the actual morphs may be conditioned by the names of the variable they modify. For example, in Latin the person suffixes are different in the passive. Since there is no overt marking of passive, this ensures that passive is overtly expressed even though not at the place where we expect it. We may call this *delayed exposure*. If however the conditioning morpheme is nonzero, this can lead to cumulative exposure. For example, the person endings in the Latin perfect are different from the ones in the other tenses. So, the presence of perfect person endings signals perfect in addition to the perfect morpheme itself, which is nonzero.

Chapter 4 presents an important extension of argument structure by parameters. Parameters are variables on which the meaning of an item depends other than those that are typically overtly expressed. For example, property ascriptions typically are time dependent, in which case they are also called stage-level predicates. (We avoid using the terminology since it is of no further significance here and we want to avoid any commitment to an accompanying theory of such predicates.) One is the director of a company for a certain stretch of time only. On the other hand, the time dependency hardly shows up in the form of an argument. It does matter on the other hand in expressions like `former` or `ex-`. The time variable has a different behaviour from typical argument variables simply because it is not identified by an overt property. Parameters therefore function differently.

There is a small number of roles each of which address a context variable. For time variables these are *story time*, *predication time* and *reference time*. Parameter statements link actual variables to these roles. They eventually get their values through the context. It is possible to relink variables to different roles, and this causes what is known as *sequence of tense*. This mechanism is not restricted to tense; Philippe Schlenker has observed that it also applies to person and world, while Kracht & Smith have shown that it additionally applies to location.

Chapter 5 presents a detailed study of case. Case is both a morphological property and a syntactic one. We start by outlining some morphological case systems and subjecting them to an analysis. Then we look at the way morphological case translates into syntactic case. The basic insight is that case that is not selected is actually semantic, while a case that is selected is syntactic. Whether or not a case is selected is a property of the head, and cannot be fixed a priori. This is our solution to the debate whether Finnish local cases are structural or semantic (see Vainikka and Niikanne). It is argued that selection is quite different from agreement. Selection is selection of a particular morpheme. Selection will make the semantic contribution of morpheme void. Agreement requires an agreement controller, and the semantics on the agreement controller is determined only by the properties of the controller. In this way plural agreement can still give rise to plural semantics, while selection of ablative case cancels its meaning completely. Of course, selection of plural has the same effect.

Contents

Chapter 1

The Combinatorics of Words

In this chapter we provide the first model of language and interpretation, devised by Montague. This allows us to introduce many notions that we will need in the sequel: types, categories, first–order logic, λ–calculus and model–theoretic semantics. Additionally, we shall talk about the semantic primitives. This chapter will be somewhat more formal than the ones to come.

1.1 The Ideal Language

We start our investigations into semantics by introducing the logical language and defining its interpretation. This has two aims. One is to fix the notation and some basic definitions; the other is to show how the logical languages lend themselves to a so–called *compositional* translation that Montague and many others following him sought to provide for natural language. As natural languages prove to be much more difficult in this respect it is good to start with a simple case (for example a formal language) and see how things differ in comparison to natural languages. The basic notions that always need to be defined for a language are its *syntax* and its *semantics*. The syntax spells out the well–formed expressions and the semantics tells us what the structures are about which this language speaks and how the formulae are interpreted in these structures. We begin with first–order logic (FOL). The following are the elementary symbols of first–order logic. Here,

ω denotes the set of natural numbers.

1. A denumerably infinite set Var $= \{\mathrm{x}_i : i \in \omega\}$ of variables.
2. A set Rel of relations, a set Fun of functions and a function $\Omega : \mathrm{Rel} \cup \mathrm{Fun} \to \omega$.
3. Propositional connectives: $\neg$, $\wedge$, $\vee$, $\dot{\to}$.
4. Quantifiers: $\forall$, $\exists$.
5. Equality: $\dot{=}$.

We use x, y etc. to denote variables, R, S for relations, and f, g for functions. The numbers $\Omega(R)$ and $\Omega(f)$ are the arity of the relation R and the function f, respectively. The symbol $\dot{=}$ is the formal symbol of identity; when we want to state in the metalanguage that two things are equal we use the symbol = instead. Let F stand for the set of all the above symbols, and F^* the set of finite sequences of members of F called **strings** (including the empty string denoted by ε). We use $\vec{x}$ and $\vec{y}$ (with vector arrow) as variables over strings. If $\vec{x}$ and $\vec{y}$ are strings, then $\vec{x}^\frown\vec{y}$ or simply $\vec{x}\vec{y}$ denotes the concatenation of $\vec{x}$ and $\vec{y}$. The set of strings of length n over F is denoted by F^n.

Definition 1.1 (Syntax of FOL) ***Tm*** *is the least subset of F^* satisfying the following properties.*

1. Var $\subseteq$ *Tm.*
2. *If $\Omega(f) = n$ and $\vec{x}_i \in Tm$ for $1 \le i \le n$ then $f^\frown\vec{x}_1{}^\frown \ldots {}^\frown\vec{x}_n \in Tm$.*

Tm is called the set of ***terms***. ***Fol*** *is the least subset of F^* satisfying the following requirements.*

1. *If $\Omega(R) = n$ and $\vec{x}_i \in Tm$ for all $1 \le i \le n$ then $R^\frown\vec{x}_1{}^\frown \ldots {}^\frown\vec{x}_n \in Fol$.*
2. *If $\vec{x}$ and $\vec{y}$ are in Fol then so are $\neg^\frown\vec{x}$, $\wedge^\frown\vec{x}^\frown\vec{y}$, $\vee^\frown\vec{x}^\frown\vec{y}$, and $\dot{\to}^\frown\vec{x}^\frown\vec{y}$.*
3. *If $\vec{u} \in Fol$ and $x \in$ Var then $\forall^\frown x^\frown\vec{u} \in Fol$ and $\exists^\frown x^\frown\vec{u} \in Fol$.*

In the remaining text we will use s and t as variables over terms and ϕ, ψ etc. as variables for formulae. Moreover, we will later use the notation $(\phi\wedge\chi)$ in place of $\wedge\phi\chi$. This is more in line with the usual notation. For the moment we would like to emphasize that the commonplace notation has a slight drawback. It uses auxiliary symbols, namely the brackets (and). Hence the sequence $(\varphi\wedge\psi)$ is strictly speaking not a member of F^*. Since the syntax of FOL provides a role model of the 'ideal' syntax, it is quite important that it does not use any extraneous means, be they brackets or other symbols.

Definition 1.2 *A **first–order stucture** is a pair $\langle D, I\rangle$, where D is a set and I is a function assigning to a relation R with $\Omega(R) = n$ a subset of D^n and to a function f with $\Omega(f) = n$ a function from D^n to D.*

Notice that we have two special cases, namely relations of arity zero and functions of arity zero. By definition, a relation of arity zero is a subset of D^0, which we take to be $\{\varnothing\}$. Hence there exist two such relations, $\varnothing$ and $\{\varnothing\}$. A function of arity zero is by construction interpreted by a function from D^0 to D. Since $D^0 = \{\varnothing\}$, we get that the function is uniquely identified by $I(f)(\varnothing)$. This is why these functions are also called *constants*.

A **model** is a pair consisting of a structure and an assignment. The interpretation in a structure is defined as follows. An **assignment** is a function $v : \mathrm{Var} \to D$. Formulae are evaluated in a model together with an assignment. Let $V \subseteq \mathrm{Var}$. We write $v \sim_V w$ if $v(y) = w(y)$ for all $y \in \mathrm{Var} - V$. Given an assignment, a term is interpreted by an element of D. We write $[t]^{\mathfrak{M},v}$ for the interpretation of t in the model plus the assignment v. $[t]^{\mathfrak{M},v}$ is defined as follows.

1. $[\mathtt{x}_i]^{\mathfrak{M},v} = v(\mathtt{x}_i)$.
2. If $\Omega(g) = n$ then $[gt_1 \ldots t_n]^{\mathfrak{M},v} = I(g)([t_1]^{\mathfrak{M},v}, \ldots, [t_n]^{\mathfrak{M},v})$.

The interpretation of formulae is defined by induction over the structure of the formula. A formula gets the value 1 or 0, corresponding to *true* and *false*. Hence we only have to specify when a formula gets the value 1, the other case is then also defined.

1. If $\Omega(R) = n$ then $[Rt_1 \ldots t_n]^{\mathfrak{M},v} = 1$ iff $\langle [t_1]^{\mathfrak{M},v}, \ldots, [t_n]^{\mathfrak{M},v}\rangle \in I(R)$.

2. $[\neg\phi]^{\mathfrak{M},v} = 1$ iff $[\phi]^{\mathfrak{M},v} = 0$.
3. $[\wedge\phi\chi]^{\mathfrak{M},v} = 1$ iff $[\phi]^{\mathfrak{M},v} = 1$ and $[\chi]^{\mathfrak{M},v} = 1$.
4. $[\vee\phi\chi]^{\mathfrak{M},v} = 1$ iff $[\phi]^{\mathfrak{M},v} = 1$ or $[\chi]^{\mathfrak{M},v} = 1$.
5. $[\rightarrow\phi\chi]^{\mathfrak{M},v} = 1$ iff $[\phi]^{\mathfrak{M},v} = 0$ or $[\chi]^{\mathfrak{M},v} = 1$.
6. $[\forall x\phi]^{\mathfrak{M},v} = 1$ iff for all $w \sim_x v$ we have $[\phi]^{\mathfrak{M},w} = 1$.
7. $[\exists x\phi]^{\mathfrak{M},v} = 1$ iff for some $w \sim_x v$ we have $[\phi]^{\mathfrak{M},w} = 1$.

The interpretation of certain formulae does not depend on the assignment. These are the closed formulae. To define them properly we need to talk about subformulae of a formula and occurrences of subformulae.

Definition 1.3 *Let ϕ and χ be formulae of first–order logic. ϕ is a* ***subformula*** *of χ if ϕ is a substring of χ, that is, if there exists $\vec{x}$ and $\vec{y}$ such that $\chi = \vec{x}^\frown\phi^\frown\vec{y}$. An* ***occurrence of*** *ϕ in χ is a pair $\langle\vec{x},\vec{y}\rangle$ such that $\chi = \vec{x}^\frown\phi^\frown\vec{y}$.*

Similarly, the notion of **subterm** and **subterm occurrence** are defined. Moreover, a **quantifier** is an expression of the form $\forall x$, $\exists x$, and we define the notion of a **subquantifier** and *quantifier occurrence* accordingly. We say of an occurrence $\langle\vec{x},\vec{z}\rangle$ of an expression α that it **occurs** inside $\langle\vec{y},\vec{w}\rangle$ if $\vec{y}$ is a prefix of $\vec{x}$ and $\vec{w}$ a suffix of $\vec{z}$. The reader for whom this is too much detail is assured that these definitions formalize the intuitive notion of *occurrence*.

We will use these definitions to define a rather crucial property of the language of first–order logic, namely its *unambiguity*. A language is called *unambiguous* if its expressions can be analyzed in essentially only one way. This does not mean that the expressions can only be *produced* in exactly one way using the inductive clauses; rather it means that the strings can only be structurally nalyzed in one way. For example, let R be a unary relation symbol, `f` and `g` be unary function symbols. The expression $\texttt{→∧Rx}_0\texttt{=fx}_0\texttt{1¬=gx}_0\texttt{0}$ can be produced by first assembling $\texttt{Rx}_0$ and $\texttt{=fx}_0\texttt{1}$ into $\texttt{∧Rx}_0\texttt{=fx}_0\texttt{0}$ and then $\texttt{=gx}_0\texttt{0}$ into $\texttt{¬gx}_0\texttt{0}$, and finally putting these two together. Alternatively, we can first produce $\texttt{¬g}_0\texttt{0}$. However, these two procedures do not differ for all intents and purposes. Namely, we will show that the interpretation of expressions that are formulae is unique. So, it does not depend on the way we arrived at the formula. The proof is by induction on the length of the string.

Lemma 1.4 (Unique Readability) *Let $\phi \in Fol$. Then exactly one the following cases obtains:*

1. $\phi = Rt_1 \ldots t_n$ *for some R with* $\Omega(R) = n$ *and some* $t_i \in Tm$, $1 \leq i \leq n$.
2. $\phi = \neg\chi$ *for some* $\chi \in Fol$.
3. $\phi = \wedge\chi_1\chi_2$ *for some* $\chi_1, \chi_2 \in Fol$.
4. $\phi = \vee\chi_1\chi_2$ *for some* $\chi_1, \chi_2 \in Fol$.
5. $\phi = \rightarrow\chi_1\chi_2$ *for some* $\chi_1, \chi_2 \in Fol$.
6. $\phi = \forall x\chi$ *for some* $x \in \mathrm{Var}$ *and some* $\chi \in Fol$.
7. $\phi = \exists x\chi$ *for some* $x \in \mathrm{Var}$ *and some* $\chi \in Fol$.

Moreover, in 1., the symbols R and $t_1, \ldots, t_n$, in 2., 6. and 7. the symbol χ and in the cases 3. – 5., the formulae χ_1 and χ_2 are uniquely determined. In other words, a formula is uniquely decomposable into a symbol followed by the immediate subformulae. This symbol we call the ***main symbol*** *of ϕ. Let $t \in Tm$. Then exactly one of the following cases obtain:*

1. $t = x_i$ *for some* $x_i \in \mathrm{Var}$.
2. $t = fu_1 \ldots u_n$ *for some f with* $\Omega(f) = n$ *and* $u_i \in Tm$, *for all* $1 \leq i \leq n$.

In the second case we call f the ***main symbol*** *of f.*

It is now clear why the interpretation is unique. Suppose that we want to compute $[\phi]^{\mathfrak{M},v}$. Then we look at the first symbol of ϕ. It is either a variable (and then ϕ is actually a term) or it is the main symbol of ϕ. If it is a function symbol, ϕ is a term $fu_1 \ldots u_n$ and by induction hypothesis $[u_i]^{\mathfrak{M},v}$ is uniquely defined. So, $[\phi]^{\mathfrak{M},v}$ is uniquely defined as well. If however the main symbol is not a function symbol, it is either a relation symbol or a quantifier symbol or a propositional connective. Again, by inductive hypothesis we deduce that the interpretation of ϕ is unique.

Definition 1.5 *Let ϕ be a formula, and $\mathfrak{Q} = \langle\vec{u}, \vec{v}\rangle$ a quantifier occurrence in ϕ. If $\mathfrak{x} = \langle\vec{w}, \vec{x}\rangle$ is an occurrence of the variable x_j in ϕ, we say that $\mathfrak{Q}$* ***binds*** *$\mathfrak{x}$ iff (1) $i = j$ and (2) $\mathfrak{x}$ occurs in the least subformula occurrence containing $\mathfrak{Q}$. An occurrence of a variable is* ***bound*** *if it is bound by some occurrence of a quantifier, otherwise it is* ***free****. A formula is a* ***sentence*** *if it contains no free occurrences of variables.*

Proposition 1.6 *Let ϕ be a sentence, $\mathfrak{M}$ a first–order model and v and w assignments into $\mathfrak{M}$. Then $[\phi]^{\mathfrak{M},v} = [\phi]^{\mathfrak{M},w}$. Hence we also write $[\phi]^{\mathfrak{M}}$ in place of $[\phi]^{\mathfrak{M},v}$.*

The proof is not hard; it uses the unique readability. Since it can be found in any textbook on logic, we will omit it.

1.2 Typed Lambda Calculus

The previous definition of the interpretion of expressions of predicate logic in models is for our purposes not ideal since it presents no uniform scheme. The interpretive clauses for different symbols are each different. Look for example at the difference between quantifiers and propositional connectives. Propositional connectives are interpreted as functions from truth–values to truth–values, while quantifiers are interpreted using alternative asisgnments. To change this, some extra machinery has to be introduced. On top of the predicate logic we also add the so–called λ–abstractor. It serves to define functions from simple expressions. This will introduce a slight complication to begin with since λ–calculus is a functional calculus and there is no place for relations. We will therefore start with a language that has only function symbols and show later how to introduce relations into it. The λ–calculus we are employing is strictly typed. In fact, throughout this book we shall assume that the universe is typed. However, we shall refrain from drawing explicit attention to the fact that we are using a typed universe later. We will speak of λ–calculus when in fact we mean typed λ–calculus.

Definition 1.7 *Let Bt be a set. The set of* ***types*** *over Bt is the least set Typ satisfying the following.*

1. *$Bt \subseteq Typ$.*

2. *If α and β are in Typ, so is $\alpha \to \beta$.*

*A member of Bt is called a **basic type**. A non-basic type is also called **composite**.*

The language of λ–calculus is now defined as follows. We have

1. For each type α a denumerable set of variables $V_\alpha = \{x_i^\alpha : i \in \omega\}$
2. A set Fun of functions and a function $\tau : \text{Fun} \to \text{Typ}$.
3. The lambda abstractor λ.

We use x, y and z as metavariables for variables. We say that x has type α if $x \in V_\alpha$. Let V be the union of all V_α. We use f, g as metavariables for functions. A function f has type α if $\alpha = \tau(f)$. Let $G := V \cup \text{Fun} \cup \{\lambda, (,)\}$. Since expressions of the λ–calculus are functions, we will simultaneously define the set of expressions and their type.

Definition 1.8 (Syntax of λ–Calculus) *The set Lb of well–formed expressions f of λ–calculus together with their type $\tau(f)$ is the least subset of G^* satisfying the following properties.*

1. *$V_\alpha \subseteq Lb$ for all α. Moreover, $\tau(x) := \alpha$ if $x \in V_\alpha$.*
2. *$Fun \subseteq Lb$.*
3. *If $\tau(f) = \alpha \to \beta$ and $x \in V_\alpha$ then $fx \in Lb$ and $\tau(fx) := \beta$.*
4. *If $\tau(f) = \beta$ and $\tau(x) = \alpha$ then $\lambda x(f) \in Lb$. Moreover, $\tau(\lambda x(f)) := \alpha \to \beta$.*

Let us note the following useful fact, which is proved analogously to the unique readability theorem of the previous section.

Proposition 1.9 *Let f be an expression of the λ–calculus. Then exactly one of the following obtains:*

1. *$f = x$ for some variable.*

2. *$f = \lambda x(g)$ for some variable x of type α and some function g of type $\alpha \to \beta$.*

3. *$f = gx$ for some g of type $\alpha \to \beta$ and some variable x of type α.*

We note that without the brackets this theorem is false. As an example, take the expression $\lambda \mathtt{x}_0 \mathtt{f} \mathtt{x}_0$. If we use no brackets, it can be read either as $(\lambda \mathtt{x}_0(\mathtt{f}))\mathtt{x}_0$ or as $\lambda \mathtt{x}_0(\mathtt{f}\mathtt{x}_0)$. In the future we will write $\mathtt{f}(\mathtt{x}_0)$ rather than $\mathtt{f}\mathtt{x}_0$ and $\lambda \mathtt{x}_0.\mathtt{f}$ rather than $\lambda \mathtt{x}_0(\mathtt{f})$. The idea behind the λ–abstractor is not so easy to explain. Typically it is used to bind off an argument place in a function. So one normally writes $\lambda \mathtt{x}_0.\mathtt{f}(\mathtt{x}_0)$, but we need in fact not abstract over variables that are arguments of the function. For example, $\lambda \mathtt{x}_0.\mathtt{x}_1$ is a well–formed expression. When applied to some z it gives $\mathtt{x}_1$.

Models for the typed λ–calculus can be built as follows. We start with a set D, the domain of objects. Objects have a type, since they are the denotata of functions. Therefore we have a function T from D to Typ, and this functions assigns a type to each object. With respect to that function, we put $D_\alpha := T^{-1}(\alpha)$, the set of all objects of type α. Obviously, to match this with the definition of λ–calculus we require that if $T(f) = \alpha \to \beta$ then $f \in {D_\beta}^{D_\alpha}$, where X^Y denotes the set of all functions from Y to X. What needs to be defined are the application and abstraction. Since abstraction is rather tricky and will not be needed later, we shall only deal with application here. This is simply defined as the normal application of a unary function to its argument. This is the standard model we will use. Not all expressions of the λ–calculus can be interpreted as such in a functional model; for example, the expressions $\mathtt{x}_4^2$ and $\mathtt{x}_0+\mathtt{x}_4$ are not a functions but terms. Functions are $\lambda \mathtt{x}_4.\mathtt{x}_4^2$ and $\lambda \mathtt{x}_0.\lambda \mathtt{x}_4.\mathtt{x}_0+\mathtt{x}_4$. Namely, functions are expressions in which every variable is bound by a λ–operator. The situation is parallel to the first–order case. What we need is the notion of a valuation. A **valuation** into $\mathfrak{F}$ is a function $v : V \to D$ such that $T(v(x)) = \tau(x)$. It assigns a concrete element for each variable such that the types match. Modulo a valuation, each expression denotes a function, that is, a member of D. We write $[f]^{\mathfrak{F},v}$ for the interpretation of f in the pair $\langle \mathfrak{F}, v\rangle$. It is defined inductively as follows.

1. $[x]^{\mathfrak{F},v} := v(x)$.

2. $[f]^{\mathfrak{F},v} := I(f)$.

3. $[fx]^{\mathfrak{F},v} := [f]^{\mathfrak{F},v}([x]^{\mathfrak{F},v})$.

A λ–**binder** is an expression of the form λx, where $x \in V$. Exactly as in first–order logic, we define the notion of an occurrence of a variable, an occurrence of a λ–binder and the notion of a λ–**bound** and λ–**free** occurrence of a variable. If all variables are λ–bound the expression is said to be a **function**. The following explains this terminology.

Proposition 1.10 *Let f be an expression of the λ–calculus without free occurrences of variables. Then for any model $\mathfrak{F}$ and any pair of valuations v, w, $[f]^{\mathfrak{F},v} = [f]^{\mathfrak{F},w}$. In this case $[f]^{\mathfrak{F},v} \in D$ and we write simply $[f]^{\mathfrak{F}}$ rather than $[f]^{\mathfrak{F},v}$.*

So, functions f get interpreted by elements of D, which is the domain of all functions. This is as it should be. As a concrete example we will show how to reinterpret first–order logic into a functional setting. The biggest problem is that we have no relations, and that functions are n–ary rather than unary. The latter problem is rather easy to solve. Suppose we have a binary function $f : X \times Y \to Z$. Then we can interpret this as a function $g : X \to Z^Y$ in the following way. We put $g(x) := h$, where $h : y \mapsto f(x, y)$. This process is known as *Currying* the function f. We write $\lambda\mathtt{x_0(\lambda x_1(f(x_0,x_1)))}$ for this function. Notice however that in our typing regime, the expression f is not a function, since it requires pairs of arguments rather than single arguments. However, in later stages we will ignore this subtlety although in this chapter it will prove to be of some importance. Namely, there are several ways to Curry a function of several arguments. For example, the expression $\lambda x_1\mathtt{(\lambda x_0(f(x_0,x_1)))}$ is different from $\lambda\mathtt{x_0(\lambda x_1(f(x_0,x_1)))}$. For example, take the function $f : \langle x, y\rangle \mapsto x^y$. Then

$$(1.1)\qquad \begin{aligned} \lambda\mathtt{x_0(\lambda x_1(f(x_0,x_1)))(5)(3)} &= 5^3 = 125 \\ &\neq 343 = 3^5 = \lambda\mathtt{x_1(\lambda x_0(f(x_0,x_1)))} \end{aligned}$$

The order in which the arguments are abstracted away is therefore important.

In order to install relations we introduce a distinction into two basic types: objects (e) and truth values (t). For each n–ary relation R of FOL we take a function f_R from n–sequences of objects to truth values that satisfies

$$(1.2)\qquad (\forall x_1)\ldots(\forall x_n)(f_R x_1, \ldots, x_n = 1 \leftrightarrow Rx_1 \ldots x_n)$$

(In fact, we will continue to write $Rx_1 \ldots x_n$, but it is now taken to be an expression in the functional sense. This removes unnecessary complications in the notation.)

Table 1.1: Types in FOL

Expression	Type
f	$e \to (e \to \ldots \to e)$
R	$e \to (e \to \ldots \to t)$
$=$	$e \to (e \to t)$
$\neg$	$t \to t$
$\wedge$	$t \to (t \to t)$
$\vee$	$t \to (t \to t)$
$\to$	$t \to (t \to t)$
$\forall$	$e \to (t \to t)$
$\exists$	$e \to (t \to t)$

In the next step we Curry all polyadic functions. Furthermore, after introducing the new type of object, the truth values, we can reinterpret the logical connectives as functions. Namely, we let $\neg$ be a function of type $t \to t$, and $\wedge$, $\vee$, as well as $\to$ functions of type $t \to (t \to t)$.

(1.3)

	$\neg$
1	0
0	1

$\wedge$	1	0
1	1	0
0	0	0

$\vee$	1	0
1	1	1
0	1	0

$\to$	1	0
1	1	0
0	1	1

Finally, we need to interpret the quantifiers. Quantifiers are expressions of type $e \to (t \to t)$. So, they take a variable and a formula and return a formula. The interpretation is as usual. $[\check{}\,\mathtt{x}_n\phi]^{\mathfrak{F},v} = 1$ iff for all $w \sim_{x_n} v$ we have $[\phi]^{\mathfrak{F},w} = 1$. The same is done for the existential quantifier. Notice that in ordinary logic we write $\check{}\,\mathtt{x}_n(\phi(\mathtt{x}_n))$. If $\mathtt{x}_n$ is an appropriate argument for ϕ, then it means that ϕ is of type $e \to \gamma$, and so $\phi(x)$ has the type γ, and therefore $\forall x\phi(x)$ has the type γ as well. In this way, the well–formed expressions of the λ–calculus of type t are exactly the formulae of first–order logic.

For example, take the language of ordinary arithmetic, with some basic functions summarized in Table 1.2. Then we get the following assignments of types

Table 1.2: Types in Arithmetic

Function	Type
0	e
$-$	$e \to e$
^{2}	$e \to e$
$+$	$e \to (e \to e)$
$\times$	$e \to (e \to e)$
exp	$e \to (e \to e)$
$\doteq$	$e \to (e \to t)$

Expression	Type
x^2	e
$y \doteq x + 1$	t
$\lambda x.x^2$	e
$\lambda x.x \times (y + z)$	$e \to (e \to e)$
$\forall x.x \doteq y^2 + z$	$e \to (e \to t)$
$\forall x.\exists y.x \doteq y^2$	t

We notice the following. An expression is a sentence exactly when it is of type t. For in this case, all variables have been bound successfully by a quantifier. The converse does not hold. This is shown by the second example. So the typing does not give much indication as for whether the expression is a function or a sentence. In fact, the following holds.

Proposition 1.11 *An expression of the λ–calculus for first–order logic is a formula iff it is of type t; and it is a sentence if in addition it contains no free occurrences of a variable.*

(The reader is asked to excuse the use of the term 'function' in connection with expressions of basic type.) Given expressions f and g and a variable x we write $f[g/x]$ for the result of replacing all free occurrences of x with g. This is defined only if x and g have the same type. It has to be defined with some care, since g may contain some variable y occuring free such that when g is inserted in place of x this variable finds itself in the scope of some λ–abstractor λy. To prevent this, the replacement is preceded by an operation that replaces each y bound by some

λy by another variable that does not occur neither in f nor in g. The exact details of this replacement do not matter here. We should however emphasize that much of the success of Montague semantics rests on the fact that the replacement operation does much of the variable management. Notice that $[y/x]$ is a metalanguage expression, not part of the language of the λ–calculus. It is a shorthand notation. We note that if $g = \lambda x(f)$ then $ga = f[a/x]$. This is an equation of λ–calculus that holds in all models under all interpretations. Therefore we write the symbol '=', which is distinct from the language internal equality symbol, which is written $\doteq$. We say that an equation $f = g$ is *universally valid* if for all models $\mathfrak{F}$ and all valuations v, $[f]^{\mathfrak{F},v} = [g]^{\mathfrak{F},v}$. There are more equations that are universally valid. For example, if f is a function and x is of appropriate type then

(1.4) $\lambda x(fx) = f$

This means that if we abstract x from fx, which in turn is f applied to the same variable x, then the resulting function is f. Therefore, in a λ–term a subexpression of the form $\lambda x(fx)$ may be simplified to f. This is called β–**reduction**. Furthermore, an expression of the form $\lambda x.f$ is identical to the expression $(\lambda y(f))[y/x]$ for a y not occurring free in f. Notice that these equations are *not* valid without this restriction. For example, $\lambda x.y$ is not the same function as $\lambda y.y$. Similarly, take the expression $f := \lambda y(+yx)$, then fx is the same as $+yx[x/y] = +xx$, and consequently $\lambda x(fx) = \lambda x(+xx)$, which is not the same as f. So, all these operations have to applied with care. Likewise, $\lambda x(fx) = f$ only if x is not free in f. We note that similar restrictions hold with respect to quantifiers. Namely, the formulae $\exists x\phi$ and $\exists y\phi[y/x]$ are equivalent only if y does not occur free in ϕ. Under similar conditions, also $\forall x\phi$ and $\forall y\phi[y/x]$ are equivalent.

1.3 Montague Semantics

While a language has only a restricted number of words (4000 is usually enough to know a language well), it has endless (we say, infinitely many) sentences that can in principle be understood by any speaker of that language. To explain this fact it was proposed that the meaning of a sentence is computed from the meanings of the words in exactly the way they are put together. One execution of this idea was that of Montague Grammar. In Montague Grammar, there was only one way of putting structures together, namely by forming a constituent. Consequently, there is only one way to put meanings together, and he proposed that the composition of meanings is by function application. Let us take a look at a simple sentence.

(1.5) `Peter watches Albert.`

(1.6) `Albert watches Peter.`

The verb `to watch` has two arguments: an actor and a theme. One is doing the watching (actor) and one is being watched (theme). For the present purposes we need not worry too much about the deeper meanings of the words 'actor' and 'theme'. Once we have understood that the meaning of the verb is a binary relation, named $\mathsf{watch}'(x, y)$, all we need to do is to is to see to it that the correct arguments are associated with x and y. We need not know for the purpose of the formalism what $\mathsf{watch}'(x, y)$ really means; we may at present simply decide that x is the actor and y is the theme. So, we want to have the following translations for the sentences. (Here, a is constant denoting Albert, and p a constant denoting Peter.)

(1.7) $\mathsf{watch}'(p, a)$

(1.8) $\mathsf{watch}'(a, p)$

How can this be achieved? We have two alternatives, and both have been pursued. The first is to assume that we have a constituent structure, and perform the translation on the basis of the constituent structure. The second, more ambitious project, is to assume no (or as little as possible) constituent structure and derive the constituent structure and the meaning in tandem from the string. Let us start with the first approach. The relevant structures are the following.

(1.9) `[Peter[watches Albert]]`

(1.10) `[Albert[watches Peter]]`

Now, rather than translating the verb by the open formula $\mathsf{watch}'(x, y)$, Montague used the λ–calculus to bind off the variables. Thereby the relation is turned into a function. So, this is now the official translation:

(1.11) $\mathtt{watches} \mapsto \lambda \mathtt{x}_0.\lambda \mathtt{x}_1.\mathsf{watch}'(\mathtt{x}_0,\mathtt{x}_1)$

It is assumed that when we have a constituent $[X\ Y]$, the translation of one of the two parts must be a function, and that this function is applied to whatever is the translation of the second argument. So, if X is translated by f and Y by a, then the translation of $[X\ Y]$ is simply $f(a)$. This is exactly what we get here since we translate $[X\ Y]$ by the term $(\lambda x.f(x))(a)$, and by the conventions of the λ–calculus this is equal to $f(a)$. Hence, if we translate `Albert` simply by a and `Peter` by p we get the following translations.

(1.12)
$$((\lambda y.\lambda x.\mathsf{watch}'(x,y))(a))(p)$$
$$=(\lambda x.\mathsf{watch}'(x,a))(p)$$
$$=\mathsf{watch}'(p,a)$$

(1.13)
$$((\lambda y.\lambda x.\mathsf{watch}'(x,y))(p))(a)$$
$$=(\lambda x.\mathsf{watch}'(x,p))(a)$$
$$=\mathsf{watch}'(a,p)$$

So, for these two simple sentences we have succeeded in our first goal. Notice that we would have gotten the wrong result if we had translated `watches` instead by $\lambda x.\lambda y.\mathsf{watch}'(x,y)$. In that case, the roles of actor and theme would have been reversed. This means that the process of Currying is not as innocent as it appears at first sight. Consider that we have as a denotation of the verb `to watch` simply a relation between individuals. Then which is the argument that we shall first abstract over? If is the object then the verb froms a constituent with its subject, if it is the subject then the verb forms a constituent with its object. Notice that in Montague's analysis the meaning of the verb `to watch` already *is* a function Curried in the right way, so that we know what the object of this verb is (its first argument) and what the object. This, however, is an artefact of Montague's own choice. We prefer to work with the terminology of thematic roles (actor and theme), or with grammatical relations (subject and object) to distinguish the various arguments. Given the semantics in these terms, the order in which we abstract the variables is arbitrary and needs to be fixed beforehand. Otherwise the semantics will fail to work properly.

Now let us turn to the second goal, namely to derive the translation from the string alone without any constituent structure. To obtain this translation, Montague introduces the typed λ–calculus for first–order predicate logic as defined in the last section. The basic types are e and t, although more can be introduced if needed. We note that with the typing regime introduced, a constituent $[X\ Y]$ can only be assigned a proper translation if the function, say X, is of type $\alpha \to \beta$ for some α and β, and Y is of type α. In that case, the constituent is of type β. A sentence is of type t. In our present examples, there still remain two possible constituent structures, both for (1.5) and (1.6), and they correspond to the following types.

(1.14) `[Peter [watches Albert]]`

$$[e \quad [(e \to (e \to t)) \quad e]]$$
$$=[e \quad (e \to t)]$$
$$=t$$

(1.15) `[[Peter watches] Albert]`

$$[[e \quad (e \to (e \to t)) \quad e]]$$
$$=[(e \to t) \quad e]$$
$$=t$$

We can make these ideas precise using some algebraic notions. Let F a set and $\Omega : F \to \omega$ a function. This function is called the **signature**.

Definition 1.12 *Let X be a set and Ω a signature. The set of Ω–**terms** over X is the smallest set $Tm_\Omega(X)$ satisfying*

1. *$X \subseteq Tm_\Omega$.*
2. *If $t_1, \dots, t_{\Omega(f)}$ are in Tm_Ω, so is $ft_1 \dots t_{\Omega(f)}$.*

This is the same as the definition of terms of Section 1.1.

Definition 1.13 *An Ω–**algebra** is a pair $\mathfrak{A} = \langle A, I\rangle$, where A is a set and for each $f \in F$ we have $I(f) : A^{\Omega(f)} \to A$. If $\mathfrak{B} = \langle B, J\rangle$ is another algebra and $h : A \to B$, then h is called a Ω–**homomorphism** if for all $f \in F$ and all $a_1, \dots, a_{\Omega(f)} \in A$:*

(1.16) $$h(I(f)(a_1, \dots, a_{\Omega(f)})) = J(f)(h(a_1), \dots, h(a_{\Omega(f)}))$$

In that case we write $h : \mathfrak{A} \to \mathfrak{B}$.

The set of terms can be turned into an algebra, called the *term algebra*. Namely, we put $\mathfrak{Tm}_\Omega(X) := \langle \mathrm{Tm}_\Omega(X), P\rangle$ where

(1.17) $$P(f)(t_1, \dots, t_n) := f^\frown t_1^\frown \dots {}^\frown t_n$$

We will however write f rather than $P(f)$. The term algebra has the following property. If $v : X \to A$ is a function and $\mathfrak{A} = \langle A, I\rangle$ is an Ω–algebra, then there is one and only one homomorphism $\overline{v} : \mathfrak{Tm}_\Omega(X) \to \mathfrak{A}$.

Let us take a special case, namely $F = \{\odot\}$ and the signature $\Omega(\odot) = 2$. Ω–algebras for this signature are called also **groupoids**. Moreover, the algebra of terms over Ω is called the **tree–algebra** over X. The background for this terminology is the following. Let X be the lexicon. (We write *Lex* rather than X.) Then we form binary branching constituent structures by forming tree terms over the lexicon. For example, the sentences (1.14) and (1.15) will be rendered as tree terms as follows.

(1.18) $\odot$ `Peter` $\odot$ `watches Albert`

(1.19) $\odot\odot$`Peter watches Albert`

Previously, we have written $[X\ Y]$ for the constituent formed by X and Y. Now we will write $\odot XY$, or, for readability, $(X \odot Y)$. The tree terms will be translated into strings of English words and into strings of λ–terms. The mapping into strings is rather straightforward. It is a homomorphism of groupoids. Observe that $\langle \mathrm{Lex}^*, ^\frown\rangle$ is a groupoid. Moreover, it is associative. That is to say, for all $\vec{x}$, $\vec{y}$ and $\vec{z}$ from Lex^* the following holds.

(1.20) $\vec{x}^\frown(\vec{y}^\frown\vec{z}) = (\vec{x}^\frown\vec{y})^\frown\vec{z}$

Definition 1.14 *Let* $s : \mathfrak{Tm}_\odot(Lex) \to \langle Lex^*, ^\frown\rangle$ *be defined by*

1. $s(x) := x$ *for* $x \in Lex$.
2. $s(\odot tu) := s(t)^\frown s(u)$

We call $s(t)$ *the string* ***associated*** *or* ***corresponding*** *to* t*.* t *is called an* ***analysis*** *of* $\vec{x}$ *if* $s(t) = \vec{x}$*.*

In informal terms, the mapping can be understood by deleting the symbol $\odot$ from the string representing the tree term. The translation into strings of λ–terms is not so straightforward to define. In fact, first we need to define the type of a tree term. The type function is partial; it only associates a type with a tree term if its translation is a λ–expression. To that end, let $b : \mathrm{Lex} \to \mathrm{Lb}$. This translation assigns λ–expressions to each basic lexical entry. Montague assumes that $b(v)$ is a term for each $v \in Lex$.

Definition 1.15 *The type $\tau(t)$ of a tree term t is defined as follows.*

1. $\tau(t) := T(b(t))$ *if* $t \in Lex$.
2. $\tau(\odot tu) := \begin{cases} \beta & \text{if } \tau(t) = \alpha \to \beta, \tau(u) = \alpha \\ \beta & \text{if } \tau(u) = \alpha \to \beta, \tau(t) = \alpha \\ \uparrow & \text{else} \end{cases}$

Definition 1.16 *The translation* $e : Tm_{\odot}(Lex) \to \langle Lb^*, ^\frown \rangle$ *is defined by*

1. $e(t) := b(t)$ *if* $t \in Lex$.
2. $e(\odot tu) := \begin{cases} e(u)^\frown e(t) & \text{if } \tau(t) = \alpha, \tau(u) = \alpha \to \beta \\ e(t)^\frown e(u) & \text{if } \tau(t) = \alpha \to \beta, \tau(u) = \alpha, \\ e(t)^\frown e(u) & \text{else} \end{cases}$

We are now ready to define a translation between strings of words and strings of λ–expressions.

Definition 1.17 *Let $\vec{x} \in Lex$ be a string of words and $\vec{g} \in Lb^*$ a string of λ–expressions. We say that $\vec{g}$ is a **meaning** of $\vec{x}$ if there exists a tree term t such that $s(t) = \vec{x}$ and $e(t) = \vec{g}$.*

The last definitions need some comments. We have defined the translation from tree terms to strings of λ–expressions in such a way that it returns a string of λ–terms even if the tree term has no type. This is desired, since we wish to assign a translation to every possible tree term. In this way, every string of words is assigned some string of terms; when the string of terms is in fact a single term, then it is a constituent and if it has type t then it is a sentence.

Definition 1.18 *A string of words $\vec{x}$ is a **constituent** of type α if it has a meaning that is a term of type α. $\vec{x}$ is a **sentence** if it is a constituent of type t.*

These concepts may be illustrated with our simple sentences. Some more complicated examples will follow later. Let us take the string (1.5). It is associated with two tree terms, namely (1.18) and (1.19). Indeed, if we apply the translation s, deleting the symbol $\odot$ from (1.18) and (1.19), then we get (1.5). Moreover, both have a type, and this is t. So, under both analyses the string is a sentence and therefore has a meaning. It turns out that these meanings differ; and in fact, only one of them is correct. We will see later how one can remedy this defect. The following strings are not constituents under any reading:

(1.21) `Peter Albert`

(1.22) `watches watches`

(1.23) `watches Peter watches`

On the other hand, the following are also sentences: [1]

(1.24) `Peter Albert watches.`

(1.25) `Watches Albert Peter.`

However, both can have only one meaning, namely that Peter does the watching while Albert is being watched (by Peter). We end this section by a theorem that assures us that at least on the semantic side everything is in the best possible order.

Proposition 1.19 *A string of words is a constituent of type α iff it has an analysis of type α.*

The proof is not so difficult. We prove by induction on the length of $\vec{x}$ that if $\vec{x}$ has an analysis ζ of type α, then it is a constituent of type α. The case is clear for a string of length 1. Now suppose that $\vec{x}$ is of length > 1 and that $\vec{x} = s(\zeta)$ with $\tau(t) = \alpha$. Then either t is a lexical entry of type α or it is of the form $\odot uv$, where $\tau(u) = \gamma \to \alpha$ and $\tau(v) = \gamma$ or $\tau(u) = \gamma$ and $\tau(v) = \gamma \to \alpha$. In the first case, $s(t) = s(u)^\frown s(v)$ and in the second $s(t) = s(v)^\frown s(u)$. Let the first be the case. By inductive hypothesis, $e(u)$ is a term of type $\gamma \to \alpha$ and $e(v)$ a term of type α. Hence $e(t) = e(\odot uv) = e(u)^\frown e(v)$ is a term of type α. Similarly the second case is handled. Now we show that if $\vec{x}$ is a constituent of type α, it has an analysis of type α. Again, if $\vec{x}$ has length 1 there is nothing to prove. So let $\vec{x}$ be of length > 1. $\vec{x}$ is translated into a sequence of expressions that is a type. Hence, there is a decomposition $\vec{x} = \vec{u}^\frown \vec{v}$ such that $\vec{u}$ and $\vec{v}$ are constituents and one of them,

[1] We allow ourselves to adapt the spelling (lower case/upper case, and punctuation) whenever appropriate.

say $\vec{v}$, is of type $\beta \to \alpha$ and the other, $\vec{u}$, of type β for some β. By induction hypothesis, they therefore have an analysis q (of $\vec{u}$) and p (of $\vec{v}$) of type $\beta \to \alpha$ and β, respectively. By this, $\vec{u}^\frown\vec{v}$ has the analysis $\odot pq$. So, $\vec{x}$ has an analysis of type α. This concludes the proof.

The importance of this theorem is the following: if we are just interested in knowing which sequences of words are grammatically acceptable (that is, if we are interested only in the constituents), we only have to investigate whether they have a type. The meaning we do not need to check. We are guaranteed a translation of identical type.

1.4 Directionality and Syntactic Types

Categorial grammar is actually much older than Montague semantics, but the interest in it has been fuelled enormously through the success of the latter. In particular, as we have seen at the end of the previous section, the semantical type check can be used to see whether a string is associated with a meaning in the translation. An analogous situation arises in physics, where the correctness of an equation can be checked by calculating the dimensions first, although that does not guarantee that we have a valid law of physics. Unfortunately, the same problem affects Montague semantics. It allows more strings to be sentences, and even assigns the wrong meaning to them. Therefore, the following remedy was introduced. In addition to the semantic type, a lexical entry gets a *syntactic* type. The syntactic type is a more elaborate version of the semantic type, and in particular it specifies the directionality. For we need to specify somewhere that English verbs expect their subject to the left and their object to the right.

Definition 1.20 *A **directional type** is a term over the set* $\{\backslash, /\}$. *The function* σ *is defined by*

1. $\sigma(b) := b$ *for each basic type.*
2. $\sigma(\alpha/\beta) := \sigma(\beta) \to \sigma(\alpha)$.
3. $\sigma(\alpha\backslash\beta) := \sigma(\alpha) \to \sigma(\beta)$

The function σ associates with a directional type a type in the previous sense. Now we define the notion of a lexicon.

Definition 1.21 *A **lexeme** is a triple $\lambda := \langle \vec{w}, \partial, g\rangle$ such that ∂ is a directional type and g an term of the λ–calculus of type $\sigma(\partial)$. We call $\vec{w}$ the **exponent** or **signifier**, ∂ the **syntactic type** and g the **meaning** of λ. A **lexicon** over Lex is a set M of lexemes with exponents from L. $\vec{w}$ is **monomorphic** in M if it is the exponent of exactly one lexeme; otherwise it is **polymorphic**.*

(Notice that exponents may be composite entities, for example strings over a fixed alphabet.) Now we have introduced a third layer, namely that of the directional types. Let us see how this gets the facts right. For our example, (1.1) and (1.2) we need the following lexicon:

Exponent	SType	Meaning
`Peter`	e	p
`Albert`	e	a
`watches`	$(e\backslash t)/e$	$\lambda x.\lambda y.\mathsf{watch}'(y, x)$

Definition 1.22 *Let u be a tree term over the set of directional types. Then a type $\delta(u)$ is associated to u in the following way:*

1. *$\delta(\partial) := \partial$ if ∂ is a type.*
2. *$\delta(\odot\vec{x}\vec{y}) := \beta$ if $\delta(\vec{x}) = \beta/\gamma$ and $\delta(\vec{y}) = \gamma$ for some γ.*
3. *$\delta(\odot\vec{x}\vec{y}) := \beta$ if $\delta(\vec{x}) = \gamma$ and $\delta(\vec{y}) = \gamma\backslash\beta$ for some γ.*
4. *$\delta(u)$ is undefined else.*

Now the machinery of Section 1.3 is applied to the present context. A string of words has several analyses. An analysis has a meaning if it has a type. This is then the meaning associated with the analysis. Let us see how this works.

(1.26) `[Peter [watches Albert]]`

$\odot\, e \;\; \odot\, e\backslash t)/e \;\; e$

$= \odot\, e \;\; e\backslash t$

$= t$

(1.27) `[[Peter watches] Albert]`

$\odot \odot e \;\; (e\backslash t)/e \;\; e$

$= \uparrow$

Now only the correct bracketing is assigned a type, and it corresponds to the right meaning of the sentence. The second analysis is successfully blocked by the directional type system. The verb wants its first argument to the right. Since there is nothing to its right, no type is assigned. The analysis is blocked.

We may note that certain basic syntactic types that are commonly known are now analyzed as rather complex types, such as the intransitive verb (t/e), the transitive verb ($(e\backslash t)/e$) and the proper name (e). For other categories we still have to find an appropriate analysis. Nouns, for example, are analyzed in the same way as intransitive verbs. This follows from their semantics, which is usually a unary predicate. For example, `man` has the meaning $\lambda x.\mathsf{man}'(x)$, which is of type $e \to t$. Therefore the syntactic type is either $e\backslash t$ or t/e. We choose the latter. There is a certain measure arbitrariness involved in assigning syntactic types to nouns, since even though the type is composite, it does not combine with anything to form a sentence. Observe that it could combine with a proper name, since it is in fact of type e. For this reason and others, Montague in fact assigns an even higher type to proper names. In English, adjective come before the noun so their syntactic type is consequently $(t/e)/(t/e)$. To give an example, `red` is assigned the meaning

(1.28) $\lambda\mathcal{P}.\lambda x.\mathsf{red}'(x) \wedge \mathcal{P}(x)$

(The following convention will be used. Lower case Roman letters will denote individual variables while calligraphic letter such as $\mathcal{P}$ and $\mathcal{Q}$ will denote variables of higher type. In this case, $\mathcal{P}$ is of type $e \to t$. The reader may check that the expression is indeed of type $(e \to t) \to (e \to t)$.) So, the expression `red man`, which is of type e/t, is given the following meaning.

(1.29)
$$\begin{aligned}\lambda\mathcal{P}.\lambda x.\mathsf{red}'(x) \wedge \mathcal{P}(x))(\lambda x.\mathsf{man}'(x)) &= \lambda x.\mathsf{red}'(x) \wedge (\lambda x.\mathsf{man}'(x))(x) \\ &= \lambda x.\mathsf{red}'(x) \wedge \mathsf{man}'(x)\end{aligned}$$

Quantifiers and determiners are a little bit tricky. Without going into detail we note that one must distinguish between quantifiers for subject noun phrases and quantifiers for object noun phrases of transitive verbs, and likewise for determiners. Namely, subject quantifiers have the following entries:

(1.30)

EXPONENT	STYPE	MEANING
`every`	$(t/(e\backslash t))/(t/e)$	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\forall x.\mathcal{P}(x) \rightarrow \mathcal{Q}(x)$
`some`	$(t/(e\backslash t))/(t/e)$	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\exists x.\mathcal{P}(x) \wedge \mathcal{Q}(x)$
`the`	$(t/(e\backslash t))/(t/e)$	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\exists x.\mathcal{P}(x) \wedge \mathcal{Q}(x) \wedge$ $(\forall y)(\mathcal{P}(y) \wedge \mathcal{Q}(y) \rightarrow y \doteq x)$

The object quantifiers are different; here, $\mathcal{P}$ is a variable of type $e \rightarrow t$ and $\mathcal{Q}$ a variable of type $e \rightarrow (e \rightarrow t)$. The resulting type is of type $e \rightarrow t$ (an intransitive verb). Hence, the object quantifiers have the semantic type $(e \rightarrow t) \rightarrow ((e \rightarrow (e \rightarrow t)) \rightarrow (e \rightarrow t))$.

(1.31)

EXPONENT	STYPE	MEANING
`every`	$(((e\backslash t)/e)\backslash(t/e))/(t/e)$	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\lambda y.\forall x.\mathcal{P}(x) \rightarrow \mathcal{Q}(x)(y)$
`some`	$(((e\backslash t)/e)\backslash(t/e))/(t/e)$	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\lambda y.\exists x.\mathcal{P}(x) \wedge \mathcal{Q}(x)(y)$
`the`	$(((e\backslash t)/e)\backslash(t/e))/(t/e)$	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\lambda y.\exists x.\mathcal{P}(x) \wedge \mathcal{Q}(x)(y) \wedge$ $(\forall z)(\mathcal{P}(z) \wedge \mathcal{Q}(z) \rightarrow z \doteq x)$

In fact, we may also translate the indefinite article `a(n)` in the same way as `some`, but we stay here with the quantifiers. To provide an example, take the sentence

(1.32) `Every man watches some red cow.`

Let us first do a categorial analysis to see whether the sentence is grammatical.

(1.33)
$$\begin{array}{cccccc}
\texttt{every} & \texttt{man} & \texttt{watches} & \texttt{some} & \texttt{red} & \texttt{cow} \\
(t/(t/e))/(e/t) & (t/e) & (e\backslash t)/e & (((e\backslash t)/e)\backslash(t/e))/(e/t) & (t/e)/(t/e) & (t/e) \\
\hline
\multicolumn{2}{c}{t/(t/e)} & & & \multicolumn{2}{c}{t/e} \\
\hline
& & & \multicolumn{3}{c}{((e\backslash t)/e)\backslash(t/e)} \\
\hline
& & \multicolumn{3}{c}{t/e} & \\
\hline
\multicolumn{5}{c}{t} &
\end{array}$$

So, the analysis indeed leads to a grammatical sentence and we should have a corresponding meaning. It is computed along the structure as follows.

(1.34)

Exponent	Meaning
every	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\forall x.\mathcal{P}(x) \to \mathcal{Q}(x)$
man	$\lambda x.\mathsf{man}'(x)$
every man	$\lambda\mathcal{Q}.\forall x.\mathsf{man}'(x) \to \mathcal{Q}(x)$
red	$\lambda\mathcal{P}.\lambda x.\mathsf{red}'(x) \wedge \mathcal{P}(x)$
cow	$\lambda x.\mathsf{cow}'(x)$
red cow	$\lambda x.\mathsf{red}'(x) \wedge \mathsf{cow}'(x)$
some	$\lambda\mathcal{P}.\lambda\mathcal{Q}.\lambda y.\exists x.\mathcal{P}(x) \wedge \mathcal{Q}(x)(y)$
some red cow	$\lambda\mathcal{Q}.\lambda y.\exists x.\mathsf{red}'(x) \wedge \mathsf{cow}'(x) \wedge \mathcal{Q}(x)(y)$
watches	$\lambda x.\lambda y.\mathsf{watch}'(y, x)$
watches every	$\lambda y.\exists x'(\mathsf{red}'(x') \wedge \mathsf{cow}'(x')$
	$\wedge(\lambda x.\lambda y.\mathsf{watch}'(y, x))(x')(y))$
red cow	$= \lambda y.\exists x'(\mathsf{red}'(x') \wedge \mathsf{cow}'(x')$
	$\wedge\mathsf{watch}'(y, x'))$
(1.32)	$\forall x.\mathsf{man}'(x) \to \lambda y.\exists x'(\mathsf{red}'(x') \wedge \mathsf{cow}'(x')$
	$\wedge\mathsf{watch}'(y, x'))(x)$
	$= \forall x.\mathsf{man}'(x) \to \exists x'(\mathsf{red}'(x') \wedge \mathsf{cow}'(x')$
	$\wedge\mathsf{watch}'(x, x'))$

This is the only possible analysis, and hence the only meaning that can be associated with this sentence. Likewise, the sentence (1.35) can only mean (1.36) and not (1.37).

(1.35) `Some red cow watches every man.`

(1.36) $\exists x.\mathsf{red}'(x) \wedge \mathsf{cow}'(x) \wedge (\forall y.\mathsf{man}'(y) \to \mathsf{watch}'(x, y))$

(1.37) $\forall y.\mathsf{man}'(y) \to (\exists x.\mathsf{red}'(x) \wedge \mathsf{cow}'(x) \wedge \mathsf{watch}'(x, y))$

This is contrary to fact and Montague has actually proposed a solution, but we will defer a discussion of that solution. The syntactic facts of English are therefore derived by a combination of two things: an assignment of meanings to lexical items and an assignment of directionality to each use of $\to$ in the type. We have already observed that this leads to certain arbitrariness in the system but let us now note what the calculus so far achieves. First of all, we get the word order and constituent structure of basic English sentences right; we have S(VO) structure, with adjectives before the noun, determiners before the noun phrase. If we change the basic word order we just have to assign a different directionality of the syntactic type assignment in the verb. Thus the following basic word orders can be accounted for:

$$(1.38)\qquad \begin{array}{lcl} \text{SVO} & : & (e\backslash t)/e \\ \text{SOV} & : & e\backslash(e\backslash t) \\ \text{OVS} & : & e\backslash(t/e) \\ \text{OSV} & : & (t/e)/e \end{array}$$

To be precise, we also need to adapt the type assignment to NP in the following way. If the object is before the verb, it gets the type V_1/V_2, where V_1 is the type of an intransitive verb and V_2 the type of a transitive verb. If the object is after the verb it gets the type $V_2\backslash V_1$. Likewise the subject gets the type V_1/V_0 (where $V_0 := t$) if it precedes the verb and $V_1\backslash V_0$ if it follows the verb. We note in passing that this way of handling the word orders shows that the directional slashes introduce a redundancy: the word order regime is displayed once on the type of the verb and the second time on the noun phrases. The reader may verify that we can set the word order on the verbs in any way we want and get the surface order right by giving the NPs the correct syntactic type.

The remaining word orders, OSV and VSO, have to be handled differently. The system does not allow to treat OV as a constituent, therefore the constituents are SV and VS. A way to repair this is therefore to change the Currying of the verb. Suppose namely that instead of the previous one we choose the following meaning.

$$(1.39)\qquad \texttt{watch} \mapsto \lambda x.\lambda y.\mathsf{watch}'(x, y)$$

Now the first argument discharged is the subject, the second the object. We will not go into the details. Remains to treat languages in which the word order is free. Here we may work with an additional constructor for syntactic types, which we denote by $\multimap$. The syntactic type $\alpha \multimap \beta$ may either compose to the right with α or to the left, that is to say, it is ambiguous between $\alpha\backslash\beta$ and β/α. Combining the possibilities of assigning a directionality (with '$/$'and '$\backslash$') and leaving it unspecified (using '$\multimap$') we can account for a broad range of word order possibilities. For example, if the transitive verb is assigned the syntactic type $e \multimap (e \multimap t)$, the possibilities SVO, SOV, OVS and VOS are simultaneously grammatical. This is a good model for Sanskrit (see [37]). That the word orders OSV and VSO are excluded, is still problematic. In Latin these word orders are perfectly acceptable.

1.5 Categorial Grammar

In the previous section we already touched the problematic issues of Montague grammar. One problematic aspect is that certain sentence structures cannot be treated canonically. Another is that for certain sentences we do not get all the meanings. Yet another problem is that the typing regime is at certain times too strict. Consider the adjectives. An adjective is a modifier of a noun. Nouns are normally one place predicates. However, there exist nouns which are relational, such as `father`, `teacher` and so on. We may for example translate the relational noun `teacher` simply by $\lambda x.\lambda y.\mathsf{teach}'(y, x)$, which is of type $e \to (e \to t)$. For the nonrelational noun `teacher` we may propose $\lambda y.\exists x.\mathsf{teach}'(y, x)$, which is of type $e \to t$. This means already that the relation between natural syntactic categories (such as 'nouns') and the semantic categories is quite difficult. On the one hand, the category $e \to t$ can be that of intransitive verbs and nouns, on the other hand a noun can be of type $e \to t$ as well as $e \to (e \to t)$. That means, a natural category can have several semantic categories and a semantic category can have several syntactic categories as equivalent. Moreover, a single exponent can have several types. This means that the calculus generates meanings for ungrammatical sentences, such as

(1.40) *`Some man man.`

The reason is that the type assignment does not distinguish between intransitive verbs and common nouns. (Not even the syntactic types are distinct!) [2] It also means that it cannot assign meanings to grammatical sentences such as

(1.41) `Some teacher of the man watches every cow.`

Now adjectives may modify nouns irrespective of whether they are relational or not. To fix this, Peter Geach [35] has proposed a rule that 'lifts' the type of an adjective (and other relevant categories) automatically. The general rule is the following.

Any exponent of type $\alpha \to \beta$ is also of type $(\gamma \to \alpha) \to (\gamma \to \beta)$. In

[2]Montague fixed that problem by assuming two kind kinds of type constructors.

the directional type system, the following liftings are possible:

$$\begin{array}{rcl} \alpha/\beta & \rightsquigarrow & (\alpha/\gamma)/(\beta/\gamma) \\ \alpha/\beta & \rightsquigarrow & (\gamma\backslash\alpha)/(\gamma\backslash\beta) \\ \beta\backslash\alpha & \rightsquigarrow & (\beta/\gamma)\backslash(\alpha/\gamma) \\ \beta\backslash\alpha & \rightsquigarrow & (\gamma\backslash\beta)\backslash(\gamma\backslash\alpha) \end{array}$$

The idea behind Geach's rule is that if instead of α we supply an α 'lacking' a γ then as a result we get a β that also lacks a γ. There is a corresponding rule as to how to transform the meaning of the item, which goes as follows.

(1.42) $\lambda y.f \rightsquigarrow \lambda \mathcal{P}.\lambda x.f[\mathcal{P}(x)/y]$

Here, y is of type α, $\mathcal{P}$ of type $\gamma \to \alpha$ and x of type γ. As is easily checked, this expression is well–typed. For example, `red` has the meaning $\lambda\mathcal{P}.\lambda x.\mathsf{red}'(x) \wedge \mathcal{P}(x)$. Raising it to a modifier of relational nouns it gets the meaning

(1.43) $\lambda \mathcal{Q}.\lambda y.\lambda x.(\mathsf{red}'(x) \wedge \mathcal{P}(x))[\mathcal{Q}(y)/\mathcal{P}]$
$= \lambda \mathcal{Q}.\lambda y.\lambda x.(\mathsf{red}'(x) \wedge \mathcal{Q}(y)(x))$

Incidentally, it is possible to extend this analysis also to the quantifiers. Starting from a subject quantifier we can derive all other quantifiers by type–raising. The rule that is necessary to accomplish this is a little bit more general than Geach's rule.

We end this survey of problems by turning our attention to logical words such as `and`, `or` and `not`. Especially the first two have very liberal syntax. Almost every pair of constituents of identical type can be conjoined by `and` and `or`.

(1.44) `cats and dogs`, `eat and drink`, `red and blue`, `in and out`

(1.45) `cats or dogs`, `eat or drink`, `red or blue`, `in or out`

The word `not` is rather restricted in its syntax, for comparison. There are in addition coordinated constructions where the two coordinated items are not of the same type

(1.46) `handsome and of good manners`,
`a Republican and proud of it`

With the exception of the latter example, the following suggestion gives satisfactory results. We propose for `and` and `or` that they have any type of the form $(\alpha\backslash\alpha)/\alpha$, where α is a type. However, it is not so straightforward to associate a canonical meaning that fits this description. We cannot simply write $\lambda x.\lambda y.(x \wedge y)$, since this would require x and y to be of type t. We will pick up that theme in Section 3.5.

Another way to extend the coverage of categorial grammar has been proposed by Mark Steedman in [90] and [91]. In his view, what needs to be generalized is not the type assignment but the mode of combining the meanings. His model is called *combinatorial categorial grammar* (CCG), since its essential novelty is the use of *combinators*.

Definition 1.23 *A* ***combinator*** *is a λ–term without any function constants.*

A combinator is therefore an expression of the λ–calculus that is made of the variables and the λ–abstractor alone. Here are some examples of combinators.

$$(1.47)\qquad \begin{array}{ll} \mathbf{L} : \lambda x.\lambda y.yx & \mathbf{R} : \lambda x.\lambda y.xy \\ \mathbf{F} : \lambda x.\lambda y.\lambda z.y(xz) & \mathbf{B} : \lambda x.\lambda y.\lambda z.x(yz) \end{array}$$

We will assume that the combinators are in fact *not* typed. That is to say, the occurring variables are not variables of the usual sort. Rather, they denote typed variables of *any* type. However, there is a condition that the resulting expression is well–typed. For example, in **L** the type of y must be of the form $\alpha \to \beta$, where α is the type of x. We will use the combinators here to allow for different combinations of meaning. For example, **L** is simply left–application, while **R** is right application; **F** and **B** are forward and backward function composition. Namely, it is checked that

$$(1.48)\qquad \begin{array}{llll} \mathbf{L}fg & = (\lambda x.\lambda y.yx)fg & = (\lambda y.yf)g & = gf \\ \mathbf{R}fg & = (\lambda x.\lambda y.xy)fg & = (\lambda y.fy)g & = fg \\ \mathbf{B}fg & = (\lambda x.\lambda y.\lambda z.y(xz))fg & = (\lambda y.\lambda z.y(fz))g & = \lambda z.g(f(z)) \\ \mathbf{F}fg & = (\lambda x.\lambda y.\lambda z.x(yz))fg & = (\lambda y.\lambda z.f(yz))g & = \lambda z.f(g(z)) \end{array}$$

The principal motivation behind CCG is the fact that in order to do gapping in categorial grammar one needs to allow for the combination of incomplete categories. That is to say, one has to allow things to combine into a constituent that would under 'normal' circumstances not be a constituent at all. An example is the sentence (1.49), derived from (1.50) by gapping.

(1.49) `Peter buys Albert a car and Sally a motor-bike.`

(1.50) `Peter buys Albert a car. And [Peter buys] Sally a motor-bike.`

We would ideally to assign to the sentence (1.49) the constituent structure (1.51), while the sentences of (1.50) should have the structure (1.52):

(1.51) `Peter buys [[Albert a car] and [Sally a motor-bike.]]`

(1.52) `Peter [[buys Albert] a car.] And Peter [[buys Sally] a motor-bike].`

It is however difficult to have it both ways. Therefore, Steedman allows (1.27) to have several constituent structures, of which one is as good as the other. Moreover, any of these alternative sentence structures will have the same meaning, so that the extra rules will not extend the meanings of the sentences but rather their syntactic possibilities. Let us see how this is achieved. We consider the combinator **F**. Let f, g and h be of type $\beta \to \gamma$, $\alpha \to \beta$ and α. Then the expression $f(gh)$ is of type γ. The expression $(fg)h$ has no type. However, the expression $(\mathbf{F}fg)h$ also has type γ, and moreover it is equal to $f(gh)$. So the combinator allows to shift the brackets of the structure from right associative $(f(gh))$ to left associative $((fg)h)$. This is exactly how we can shift from the constituent analysis (1.52) to (1.51). In fact, both (1.51) and (1.52) coexist; any mode of composition is allowed, provided it meets certain constraints. These are as follows.

Definition 1.24 *Let α be a type. β is an* ***argument*** *of α if either (i) $\alpha = \beta \to \gamma$ for some γ or (ii) $\alpha = \gamma \to \delta$ for some γ and δ and β is an argument of δ. β is the* ***first*** *argument of $\beta \to \gamma$. If β is the nth argument of α, then it is the $n+1$st argument of $\gamma \to \alpha$. β is a* ***result*** *of α if either (i) $\alpha = \gamma \to \beta$ for some γ or (ii) $\alpha = \gamma \to \delta$ for some γ and δ and β is a* ***result*** *of δ.*

Not all compbinators are admitted, otherwise just any string combination rule would be possible. Therefore, some restrictions must be put. The following generalized combinators are admitted:

(1.53) $((\mathbf{F}^n\gamma[\alpha])\beta \to \alpha) = \gamma[\beta]$

(1.54) $((\mathbf{B}^n\beta \to \alpha)\gamma[\alpha]) = \gamma[\beta]$

Here, $\gamma[\alpha]$ denotes a type γ with a fixed occurrence of the type α. The combinator allows to combine $\gamma[\alpha]$ with $\beta \to \alpha$. It returns the type γ ,with the specific occurrence of α replaced by β. The proviso in both cases is that the fixed occurrence of α is that of the nth argument of γ. A special case is $\gamma = \alpha \to \delta$. Clearly, α is an argument of γ, and the result of replacing α by β is $\beta \to \delta$. Hence, the combinators **F** and **B** are cases $n = 1$ of the combinators $\mathbf{F}^n$ and $\mathbf{B}^n$.

In addition to the new modes of type composition we also need to consider the new syntactic possibilities that arise. Steedman assumes in fact that not all possible combinations of assigning directions to the arguments are allowed. In this way he correlates the basic word order of a language with the directionality of gapping. (See the discussion in Section 2.5.)

1.6 Basic Ontology or: The Basic Types

Many syntactic facts can be accounted for if the semantic analysis is more subtle and allows for a more fine grained semantic typology. This has been the direction taken among others by Glyn Morrill (see [68]). In this section we will investigate this possibility and take the chance to introduce the basic categories with which we will work in the subsequent chapters. The foremost problem so far is that of adverbs. There are several types of adverbs: sentential adverbs, manner adverbs, temporal adverbs and so on. Sentential adverbs are `sf hopefully`, `luckily`, `sf possibly`, `sf certainly` and others. They typically appear at the beginning of the sentence. Manner adverbs are close to the verb, while temporal adverbs appear in inermediate position. But these are just rules of thumb. For an enlightning discussion see Bouchard [15]. The semantical translation employed so far has no place for adverbial modification. For this (and many other) reasons it has been proposed to introduce **events** (see Parsons [74]). It is perhaps hard to describe what an event is. For the moment it may be enough to say that verbs denote events, though later we shall become more specific on this point. For example, the sentence (1.55) says that there is an event of walking, and that the one doing the walking is John.

(1.55) `John walks`.

To introduce the actants of verbs we actually assume some basic thematic roles. In our example, John is actor of the event of walking. We will not discuss these notions further. We will not concern ourselves with the question of what it means to be the actor of an event. Following the mainstream (see Cook [20]) we will assume the following roles: **actor**, **theme**, **goal**, **instrument**, **beneficiary** and **location**. Verbs select up to three arguments, all other arguments are freely added. The verb `walk` has the following bare semantics:

(1.56) $\mathsf{walk}'(e) \wedge \mathsf{act}'(e) \doteq x$

This says in informal terms that we have an event e of walking and an actor x of that event. Notice that we assume for each thematic role ϑ a function that gives for input e the object bearing the role ϑ. If that is group (see below) then $\vartheta(e)$ is the entire group. The function ϑ may also be partial. For it may often enough be the case that there is no bearer of a particular θ–role. For example, in (1.56) there is no beneficiary. If that is so, the function ben' is undefined. If we want to incorporate events into Montague semantics we need to introduce a new type which denotes events. Let us denote it by ε. To distinguish it from the type e of objects, we prefer to rename the latter into o. Then intransitive verbs have the type $o \to (\varepsilon \to t)$, transitive verbs the type $o \to o \to (\varepsilon \to t)$.

(1.57)

Exponent	SType	Meaning
`walk`	$o\backslash(t/\varepsilon)$	$\lambda x.\lambda e.\mathsf{walk}'(e) \wedge \mathsf{act}'(e) \doteq x$
`watch`	$(o\backslash(t/\varepsilon))/o$	$\lambda x.\lambda y.\lambda e.\mathsf{watch}'(e) \wedge \mathsf{act}'(e) \doteq y$ $\wedge \mathsf{thm}'(e) \doteq y$
`slowly`	$(t/\varepsilon) \multimap (t/\varepsilon)$	$\lambda\Omega.\lambda e.\mathsf{slow}'(e) \wedge \Omega(e)$

There is a slight problem in this translation, namely that sentences will be of type t/ε. At some point we need to bind the event variable by a quantifier. The assumption here is that this can be done by adverbials (`always`, `sometimes`, `under certain circumstances` and so on).

The adverbials that bind the event variable are usually time adverbials. It has therefore been proposed that tense also is to be treated as an adverbial —- and not vice versa. The reason for this is that tense is usually interpreted as existential, while the adverbials have a much wider range of possibilities. We will have little to say about these matters in what is to come, though.

In order to incorporate tense into the semantics and syntax, we need to introduce **locations**. There are two types of locations, **spatial** and **temporal** locations. An event usually lives in a certain region of space and time. We call it the **event-**

location (for the spatial region) and the **event-time**.

(1.58) `Albert was picking plums in the garden of the neighbours last night.`

(1.59) means that there is an event of plum–picking of which Albert was the actor, whose spatial location was the garden of the neighbours and which took place last night. In fact, the expression `garden of the neighbours` is not a location, rather the expression `in the garden of the neighbours`. The analysis is roughly as follows. The location of the event was included in the location of the garden of the neighbours. The expression `last night`, however, truly denotes a time region. We will not be interested much in what shape the space–time region of an event has, but we will generally assume that it is path–connected. This means for the event–time that it is an interval. This interval has a beginning and an end point. These are quite important elements to properly analyze sentences involving a transformation.

(1.59) `Alfred hammered the block into an axe blade.`

(1.60) `Albert flew the plane into Berlin.`

(1.61) `The professor came out of the bar.`

Furthermore, some locational adverbials refer to intervals in between the end points. Examples are the following.

(1.62) `She came in through the bathroom window.`

(1.63) `Albert flew the plane over the Alps.`

For example, (1.63) means that there is an interval contained in the event time of Albert's flying where the airplane was above the Alps. More about these matters in Chapter 5.

It would take too much to review the various proposals concerning the integration of tense and time into semantics and syntax (see Reichenbach, ter Meulen, Hornstein [48]). In fact, we will in the sequel use a rather primitive semantics for time, namely based on time points and intervals.

Another new semantic type is that of a **group** or **collection**. This is needed to account for noun phrases. For example, take the sentence

(1.64) `Three boys are playing with 30 marbles.`

One way of interpreting (1.37) is as follows. There were three boys in total and thirty marbles in total, and the boys were playing with the marbles. We will represent this as follows. (Upper case letters denote groups, $\sharp X$ is the cardinality of the set or group X. We shall also write $(\forall x \in X)\varphi(x)$ or $\forall x \in X.\varphi(x)$ in place of $\forall x.x \in X \to \varphi(x)$ and $(\exists x \in X)\varphi(x)$ or $\exists x \in X.\varphi(x)$ in place of $\exists x.x \in X \wedge \varphi(x)$.)

(1.65) $\exists e.\exists X.\exists Y.\mathsf{play}'(e) \wedge \mathsf{act}'(e) \doteq X \wedge (\forall x \in X.\mathsf{boy}'(x)) \wedge \mathsf{inst}'(e) \doteq Y \wedge (\forall y \in Y.\mathsf{marble}'(y)) \wedge \sharp X \doteq 3 \wedge \sharp Y \doteq 30$

We will not investigate further what it means that three boys play with thirty marbles; we assume that the sentence is simply vague on this point. Notice however that there are a number of things that are quite consistent in language. Plural nouns denote groups of things of the quality denoted by the noun. Therefore, `boys` means $\lambda X.\forall x \in X.\mathsf{boy}'(x)$ and not $\lambda X.\mathsf{boy}'(X)$. The latter would rather be a group that is a boy. So, adjectives modifying a plural noun are interpreted distributively. For example, `six large people` means a group of six people of which each is large, while `a large group of people` means a group of people such that there are many people in that group.

(1.66) $\sharp X \doteq 6 \wedge (\forall x)(x \in X \to (\mathsf{person}'(x) \wedge \mathsf{large}'(x)))$

(1.67) $\mathsf{large}'(X) \wedge (\forall x)(x \in X \to \mathsf{person}'(x))$

Therefore, we need to distinguish an adjective that modifies a group–variable from one that modifies an individual variable. These are somehow difficult concepts, since they are in fact aspectual rather than intrinsic properties. For on the one hand the group is also an individual (at least from the logical point of view), so we must distinguish the group acting as a unity from the group acting distributively, namely by virtue of each member of the group acting individually, not together with the other members of the group. In this respect note the difference between (1.64) and (1.68), which rather means (1.69).

(1.68) `Three boys are playing with 30 marbles each.`

(1.69) $\exists e.\mathsf{play}'(e) \wedge \exists X.\mathsf{act}'(e) \doteq X \wedge \sharp X \doteq 3 \wedge (\forall x \in X.\mathsf{boy}'(x)) \wedge (\forall x \in X.\exists Y.\mathsf{instrument}'(e) \doteq Y \wedge \sharp Y \doteq 30 \wedge (\forall y \in Y.\mathsf{marble}'(y)))$

While the sentence (1.64) has two groups as actants, (1.68) has four groups, a group of boys and three groups of marbles, one for each boy. Thus (1.69) shows the typical effect of one quantifier taking scope of another. (1.64) arguably has the reading (1.69), too. In that case the verb is said to interpret the subject distributively. In principle, also the object may be taken distributively. In that case, each boy is playing with each marble. Moreover, there is an additional problem with the event variable in the sense that if we read some of the noun phrases distributively we end up with a group of events rather than a single event. For example, (1.64) can mean that there are three events, each one containing a boy who is playing with 30 marbles. The distinction between one and several events is of course a subtle one; it depends on many factors, such as whether the events are far enough from each other, whether there is a sense of togetherness in the playing and so on. Furthermore, the way in which a group is made up from individuals plus the meaning of the verb may show subtle interaction as for how the sentence is understood.

(1.70) `John and Sue married.`

(1.71) `John and Sue and Peter and Ann married.`

(1.72) `Last night, everyone danced with everyone.`

We assume that there is a use of `and` that forms groups. So, (1.70) can mean that John married someone and Sue married someone. It can also mean that a group consisting of John and Sue married. As is fixed in the meaning of `marry`, a group that marries consists of two individuals, a woman and a man. [3] Therefore, (1.71), offers several readings depending on whether we interpret the various occurrences of `and`, the group–forming `and` or the logical `and`. For example, we might interpret the first and the third use of `and` in the group–forming sense and the second in the logical way. Then we have two events of marrying, one where John and Sue married, and one where Peter and Ann married. Of course, we may also have one event in which both marriages take place. Similarly, (1.72) is by normal standards understood as: every man danced with every woman. This may or may not be a consequence of the meaning of the verb `dance`, but anyhow it is difficult to account for such readings in a strict way. For our purposes, groups are **sets**. (For a defense of this view see Landmann [61] and [62].) To speak of sets, we need to introduce a separate type of **numbers** (functioning as cardinalities), the relational symbol $\in$ and the function $\sharp$ from sets to numbers.

Some additional machinery is needed to account for **mood**. First of all, there is a distinction between **realis** and **irrealis**, which has already been included in Montague's system through the use of **possible worlds**. His semantic types were more

[3]This may or may not be considered as part of of the meaning of the verb `to marry`. We are not taking any position on that issue here.

subtle than ours, allowing for the treatment of propositional attitudes. Namely, the sentence ((1.73) is not to be translated by (1.74).

(1.73) `Albert searches for the ideal woman.`

(1.74) $\exists x.\textsf{ideal-woman}'(x) \wedge \textsf{search}'(a, x)$

This would mean that there is someone that is an ideal woman and Albert searches for her. The problem is that the quality of being an ideal woman is pretty much up to everyone's own standards, so we should perhaps say that there is someone who Albert thinks is an ideal woman and he is searching for her. But even the existence of this person is very much in question. What we would ideally like to say is that Albert is trying to find (in some possible world) a person that he thinks is an ideal woman (for him). Possible worlds are however a very special instrument, since they produce all kinds of difficult questions (such as trans–world identity). We prefer to keep them out of discussion. Furthermore, we will henceforth ignore point–of–view attributions such as `the ideal woman`. We will make occasional use of modalities but try to avoid the deep waters.

The same applies to the illocutionary acts, **questions**, **desires**, **commands** and so on. We will make the following basic assumptions. Only main sentences can contain an illocutionary act, and — taking away obvious exceptions such as appositions — there can be only one act with one content. The act of locution takes place in a certain context, that involves (at least): the speaker, the hearer, a space–time region and a possible world. It is these coordinates to which the speaker can refer using indexical expressions. Speaker–hearer indexicals are abundant in languages, being codified for example in the verbal inflection, locational indexicals include expressions such as `here`, `there`, `now`, `later`. In a speech act, the speaker utters a sentence S. Through this act of utterance, the coordinates are fixed. Through the speech act, moreover, the speaker endows the sentences with a certain **force**. He can ask, command, assert, deny and so on. It is therefore important to distinguish the type of a **proposition** from that of an **utterance**. An utterance is the act itself, through which the proposition is endowed with the force and through which the indexicals are 'anchored' into the model. Again, we will have little to say about the exact nature of the primitives and how the interpretation is set up properly.

Chapter 2

New Semantic Structures

In this chapter we shall introduce Discourse Representation Structures (DRSs) of Kamp and Referent Systems by Vermeulen. The two will be merged into a new semantics for natural language, which is based on variable sharing by overt agreement. However, several changes will be made to the referent systems to accommodate for several special features of language. After they are introduced, we shall derive some basic properties of this semantics. We shall show how to derive $\overline{X}$-syntax and alternate constituent orders.

2.1 Problems with Montague Grammar

One of the tasks of semantics in the view of Montague is to say in which way the meaning of a complex expression is derived from the meaning of its parts. Montague solved this task by assuming that the denotata for words are particular λ-terms. In wa sense, the introduction of λ-calculus into semantics only serves one purpose: to track variables. Substitution was taken care of as well, because it is inbuilt into function application. Montague tied his approach with another requirement, namely that syntax should be a reflex of semantics. Although there is massive evidence against this view—one being that the directionality of selection simply does not follow from the semantics—it nevertheless captures an essential intuition which many syntactic theories have difficulties integrating without stipulation. It is that words cannot take more arguments than they can use in their semantic representation. However, they are often allowed to take less arguments, a fact that is not reflected at all in Montague Semantics. The problem with Montague's conception was that evident syntactic properties do not fall out of the meaning, such as directionality. Moreover, if we take a verb, say `loves`, then what is it in the *meaning* of this word that determines the object to be consumed first, and then the subject? In other words, why is it that `loves` denotes a function taking its object as first argument and only after that the subject? Why is it a function at all, and moreover why does it have to be Curried? One possible answer is that it is simply a fact of universal grammar. Language decides that the object be the first argument and the subject the second. The problem with this view is that if that is so, there is nothing to distinguish a transitive verb from an intransitive verb with two arguments. For example, there are verbs in German that take a dative object. From Montague's viewpoint this verb should have the same semantics as an ordinary transitive verb. But where is the difference between these two types of verbs reflected?

Other syntactic theories also have problems with the alignment and discharge of arguments. Take for example Government and Binding theory. The standard analysis of basic word order in German is that the verb discharges the arguments at D-structure in a canonical order. The alternative serialisations are obtained by a subsequent movement of these arguments out of the verb phrase. These facts need extensive argumentation, though, since we need to establish first of all diagnostics for the D-structure and then apply them to the sentences of German. (See Haider [42] for an exposition.) The diagnostics are usually fragile, and they are mainly based on binding theory. A similar approach can however also be taken in Montague Grammar. Assume that a sentence has in fact two structures; one is the one we see, called the **phenogrammatical structure**. The other is the one that is obtained by tracing the structure building process. This is called the **tectogrammatical structure**. This distinction only makes sense if we disconnect syntactic and semantic constituency. For then we do not need to assume that when we combine two constituents, say X and Y, and form the constituent denoted by $[X\ Y]$, the exponent of $[X\ Y]$ is necessarily the concatenation of the exponents of X and Y. Instead, we can take it to be a more complex structure. For example, if constituents are pairs of strings there are more fancy operations that can be defined, in particular an operation known as *wrapping*. Wrapping is defined as follows:

$$\begin{array}{ccccc} \langle \vec{u}, \vec{v} \rangle & \circledast & \langle \vec{x}, \vec{y} \rangle & := & \langle \vec{u}\vec{x}\vec{v}, \vec{y} \rangle \\ \vdots & & \vdots & & \vdots \\ \mathrm{X} & & \mathrm{Y} & & [\mathrm{X} \quad \mathrm{Y}] \end{array} \tag{2.1}$$

Notice that wrapping changes the order of the structures. Before wrapping we had first $\vec{u}$, then $\vec{v}$, and on the other hand first $\vec{x}$, and then $\vec{y}$, and after wrapping we have first $\vec{u}$, then $\vec{x}$, then $\vec{v}$, and finally $\vec{y}$. Such operations indeed split phenogrammatical structure and tectogrammatical structure. For example, as Calcagno [17] shows, Dutch and Swiss German can be assumed to be tectogrammatically context-free. For the phenogrammar this is false (see Huybregts [49] and Shieber [86]). The wrapping analysis gets the facts right, and corresponds quite neatly to the movement analyses proposed in transformational grammar. The split between phenogrammar and tectogrammar is needed also for reasons other than pure generative capacity. Namely, categorial grammars do not associate the correct constituent analysis for languages that have VSO or OSV word order, and they fail to give an analysis to languages in which the word orders OVSI and ISVO are legitimate, where I (= indirect object) is the first argument of the verb, O the second, and S the third. This applies as well to Steedman's system, which allows for more parses than Lambek-grammar, as we will see below.

We shall argue in this chapter that with very few exceptions one is not forced to assume a split between syntactic and semantic constituency as long as the semantics is defined in an appropriate way. There are a number of drawbacks of the wrapping analysis. The first is that it is more complicated than pure concatenation. We have two choices for syntactic theory. Either we assume strict word order and explain the free word orders to be subject to a liberation from the regime by whatever means (transformations, wrapping). Or we also declare free word order to be a basic choice and strict word order the effect of constraining the serialisation. Both approaches have their appeal. If we choose the first option then with respect to languages like German or Latin we have to give an explanation as to why it is not harder to understand a sentence where arguments do not appear in the canonical order. In transformational grammar this paradox is avoided by dropping the realist interpretation of transformations. So transformational grammar is like head grammar: the different word orders are simply the effect of choosing a different mode of combination, say concatenation versus wrapping. The simplest of all theories, however, would be one in which we would not have to choose at all between modes of combination. This would mean that different argument serialisations of

German sentences are base generated. Unfortunately, this theory is too simplistic. With respect to free word order a lot of arguments have been adduced to show that indeed German (and many other similar languages) have an underlying canonical word order (see Haider [42]). These theories are based on two main data: binding theory and focus spreading. Binding theory has undergone many revisions and therefore the evidence based on it has to be taken with a grain of salt. The data on focus spreading however stands out and needs accounting for. Nevertheless, as has been argued by Pollard and Sag in [80] (and many others), it is not necessary to assume syntactic movement in order to explain these facts. As we will see in this chapter, one can have both a canonical argument and free constituent order at the same time, without any further assumptions.

It should be said, though, that the semantics we are going to propose has not primarily been developed to account for movement, but to account for case marking and agreement morphology. There is to our knowledge no semantical theory that deals with agreement in any significant sense of the word. This may have to do with the fact that Indo-European languages do not have much of morphology left to deal with, and this has allowed Montague to account for agreement with a simple listing of cases. (Basically, English agreement morphology consists (except for pronouns and the copula `to be`) in the following: the verb carries a suffix `s` if the subject is third singular, otherwise not. Analogously a rule for plural noun phrases must be defined.) But this is a far cry from a sensible theory of case and agreement. In GB, the case assignment properties and other characteristics of a word are listed in its **argument structure**. The argument structure of a word tells us what arguments this word takes and how they must be linked into the semantics. In Montague Semantics there is only a rudimentary argument structure, namely the semantic type. The linking is taken care of by the λ-binding mechanism. Therefore, there is nothing in the system that corresponds to case assignment. The object is identified qua being the first argument to be taken by the verb and not qua having a certain case. The latter must be stipulated on top of the meaning. This has been done in Bierwisch [12]. According to Bierwisch, an argument structure is a λ-expression in which the λ-bound variables are annotated for morphological or syntactic requirements. Take for example a verb with three arguments. Its representation is something like $\lambda x.\lambda y.\lambda z.\phi(x, y, z)$. The argument structure pairs the arguments with cases. For example, the German `aussetzen` ('to expose someone to something') will give

$$(2.2) \qquad \begin{array}{cccl} \lambda x & \lambda y & \lambda z & \mathsf{expose}'(z, y, x) \\ \vdots & \vdots & \vdots & \\ \text{DAT} & \text{ACC} & \text{NOM} & \end{array}$$

When the verb discharges the arguments one by one we put a side condition on the rule that the argument must meet the requirements put on it in the argument structure of the functor. This takes care of the D-structure. For example, the first argument is marked for dative. Hence the first argument must have dative case.

The annotation of case information poses some intricate questions. The main question of course is: how do we knit the case requirements into the λ-calculus? One option is to declare the cases to be types. Then the variables are typed according to their case, and the whole machinery works as desired. Indeed, it would be possible to redo large parts of what is to follow in a version of typed λ-calculus that employs types to encode morphological properties. However, this appears to be problematic for a number of reasons. First, there are operations that change the grammatical function. One such example is the passive.

(2.3) `Johns eats a pizza.`

(2.4) `A pizza is eaten by John.`

Under a view that takes grammatical functions (or cases, for that matter) to be types, the phrase `a pizza` has two different types in (2.3) and (2.4). If types have anything to do with semantics at all passive is therefore not meaning invariant, because it changes the types of the arguments. Second, agreement involves also categories that do have semantic relevance, such as gender and number. That appears at first sight to be an argument in favour of the above view. However, at closer look there are a number of languages in which syntactic gender is not semantically motivated at all. One and the same thing can be either masculine or neuter, for example, depending on how we refer to it. A case in point are German diminutives, which are always neuter. So, `Kätzchen` is neuter, while `Katze` is feminine. Further, `Fräulein` (English 'Miss') is neuter, although it refers strictly to women! What these examples show is that in addition to a semantic gender (which in German does exist after all) we must distinguish a syntactic gender. The latter however cannot consistently be said to be part of a typing system other than a purely arbitary, nonsemantic one, which has to exist in addition to the semantic classification into gender. The third reason to disfavour the view that syntactic classes are types is that it forces us to assume as well types for case, types for grammatical functions, types for θ-roles and so on. This, however, is highly suspicious. Since typically all these categories work together in a language, we end up with a highly structured type system that has very little to do with semantics in the genuine sense of the word. We take it therefore that the view of syntactic categories as semantic types is highly implausible.

An alternative solution is to consider the annotation as a condition on constituent formation. In that case it is a morphological side condition on merge. This seems to be the view of Bierwisch. Also this option is difficult to maintain at closer look. One immediate argument against it is that it is usually the verbal complement that is the function and not the verb itself. However, if the property DAT is connected with the most immediately abstracted variable of the verb `aussetzen` ('to expose'), the restriction connected with the λ-abstracted variable is unpacked not at the constituent juncture but at a reduction step. To put that differently: even when the verb has a λ-bound variable, that does not mean that it is the function in a constituent and applies by discharging the variable. Rather, the dative object is taken to be the function; it takes the verb as an argument. Only in a later reduction step will the λ-abstracted variable be discharged. Here is an example.

(2.5)

`einer Gefahr` `aussetzen`

$$\lambda\mathcal{P}.\lambda y.\lambda z.\exists x.\mathsf{danger}'(x) \wedge \mathcal{P}(x)(y)(z) \quad (\lambda x.\lambda y.\lambda z.\mathsf{expose}'(z,y,x))$$
$$\mapsto \lambda y.\lambda z.\exists x.\mathsf{danger}'(x) \wedge \lambda x.\lambda y.\lambda z.\mathsf{expose}'(z,y,x)(y)(z)$$
$$\mapsto \lambda y.\lambda z.\exists x.\mathsf{danger}'(x) \wedge \mathsf{expose}'(z,y,x)$$

Thus, the verb is consumed through the variable $\mathcal{P}$, and nothing is written into the condition on x in the noun phrase, since x is not λ-abstracted over. When we do β-reduction, however, the function corresponding to aussetzen is applied to its argument, and only at this stage the condition that the variable be connected to a dative argument becomes active.

The merge leads to a discharge of all abstracted variables in the verb and to a reabstraction of y and z. We have seen already that this roundabout way of doing things is necessary in order to handle quantifiers. On the other hand, if a verb does not combine with an argument but an adverb, then the restrictions on constituent formation do not apply. It seems therefore that the case annotation is not simply a condition on constituent formation but rather a condition on such constituent formation in which the relevant variable is discharged. This already is a difficult point for the constituent interpretation.[1] Another problem arises with the mediation of the morphological restrictions. A noun usually has no arguments. Its argument structure is therefore empty and does not display any information as for its case. Hence, the case information must be separately annotated. An adjective, `blau` ('blue'), for example, has a unique argument, and so its meaning is something like $\lambda\mathcal{P}.\lambda x.\mathsf{blue}'(x) \wedge \mathcal{P}(x)$. Following Bierwisch, we would assume the following argument structure.

$$\begin{array}{lll} \lambda\mathcal{P} & \lambda x & \mathsf{blue}'(x) \\ \vdots & \vdots & \\ \gamma & ? & \end{array} \qquad (2.6)$$

What will we put in place of γ? Since an adjective modifies nouns of any case, γ must be left underspecified in the lexical entry. We can view the instantiation of γ as adding the morphological case. Notice that the variable x has no syntactic correlate and therefore nothing is connected with it in the argument structure. That means that the agreement suffixes of the adjective are interpreted as showing us what the argument structure is. Notice, however, that standard GB theory assumes only case to be a morphological requirement, but not other features (which are common;y referred to as ϕ-features). The agreement between adjective and modified noun are thus unexplained. With some charity, however, we can expand the proposal to state that morphological requirements can include ϕ-features. Still, there are problems. First of all, how do the conditions on a property ($\mathcal{P}$) relate with those of the object (x)? Moreover, what remains after the adjective has composed with the noun? We have to say what morphological properties the complex adjective+noun has, since it too can be the argument of another adjective. And since both adjectives must generally agree in case and ϕ-features, we are led to conclude that the morphological properties are passed up after composition. But how is that achieved? If the morphological requirements are conditions on forming constituents, there is no rule telling us what requirements the result satisfies. This type of argument structure explains only what happens if arguments are consumed once and for all, not what happens if they are modified. That this is not a trivial point is indicated by coordination. Here, the coordinated noun phrase `John and`

[1] In GB, adverbs are not allowed to intervene between the arguments and the verb. This is also the view of Bierwisch, who considers this as a model to generate the D-structure. Therefore, this argument has less force than appears at first sight. Nevertheless, if such a view is adopted it becomes less obvious how semantics and syntax are coupled, because we still need to account for the transformations, which are not always neutral with respect to meaning.

Mary shows plural agreement while John and Mary both show singular agreement.

2.2 Basic Semantic Concepts: DRT

We will depart from Montague's conception in several ways. First of all, we shall dispense completely with λ-calculus. The meaning of the word man, for example, will no longer be $\lambda x.\mathsf{man}'(x)$ but rather $\mathsf{man}'(x)$, where x is a variable. In fact, this is also the idea of DRT, where—for different reasons—the meaning of the word man is a pair $[\{x\} : \mathsf{man}'(x)]$. This is usually denoted in the form of a split box, also called a **Discourse Representation Structure** (DRS). (See [50] for an introduction to Discourse Representation Theory (DRT).)

(2.7)
$$\begin{array}{|l|}\hline x \\ \hline \mathsf{man}'(x) \\ \hline \end{array}$$

We call the upper part the **head** and the lower part the **body** of a DRS. The body contains the information of the DRS. The head contains the variables that are existentially quantified over.

Definition 2.1 *A **DRS** is a pair $[V : \Delta]$, where V is a finite set of variables and Δ a finite set of formulae or DRSs. The set of DRSs is constructed as follows.*

1. *If x is a variable then $[\{x\} : \varnothing]$ is a DRS.*
2. *If ϕ a formula then $[\varnothing : \phi]$ is a DRS.*
3. *If δ is a DRS then $[\varnothing : \delta]$ is a DRS.*
4. *If $[V_1 : \Delta_1]$ and $[V_2 : \Delta_2]$ are DRSs then so are*
   - *(a)* $[V_1 \cup V_2 : \Delta_2 \cup \Delta_2]$
   - *(b)* $\neg[V_1 : \Delta_1]$
   - *(c)* $[V_1 : \Delta_1]{:}[V_2 : \Delta_2]$
   - *(d)* $[V_1 : \Delta_1] \vee [V_2 : \Delta_2]$

We write $f \sim_V g$ if for all $y \notin V$ we have $f(y) = g(y)$. There are more constructors to form DRSs, but the ones above shall suffice for now.

Definition 2.2 *Let $\mathfrak{M} = \langle D, I\rangle$ be a first-order model, $f : Var \to D$ be an assignment and δ be a DRS. δ is* ***true*** *under the assignment f, in symbols $\langle \mathfrak{M}, f\rangle \models \delta$, if the following holds.*

1. *$\delta = [V : \Delta]$ and there exists a $g \sim_V f$ such that $\langle \mathfrak{M}, g\rangle \models \gamma$ for all $\gamma \in \Delta$.*
2. *$\delta = \neg[V : \Delta]$ and for no $g \sim_V f$ we have $\langle \mathfrak{M}, g\rangle \models \gamma$ for all $\gamma \in \Delta$.*
3. *$\delta = [V_1 : \Delta_1] \vee [V_2 : \Delta_2]$ and either there exists a $g \sim_{V_1} f$ such that $\langle \mathfrak{M}, g\rangle \models \gamma$ for all $\gamma \in \Delta_1$ or there exists a $g \sim_{V_2} f$ such that $\langle \mathfrak{M}, g\rangle \models \gamma$ for all $\gamma \in \Delta_2$.*
4. *$\delta = [V_1 : \Delta_2]{:}[V_2 : \Delta_2]$ and for all $g \sim_{V_1} f$ such that $\langle \mathfrak{M}, g\rangle \models \gamma$ for all $\gamma \in \Delta_1$ there exists a $h \sim_{V_2} g$ such that $\langle \mathfrak{M}, h\rangle \models \gamma'$ for all $\gamma' \in \Delta_2$.*

We define now the notion of accessibility and boundedness. Let $\delta = [V : \Delta]$; then δ is **immediately accessible** for every $\gamma \in \Delta$. Furthermore, in $\delta'{:}\delta''$, δ' is immediately accessible for δ'', but δ'' is not immediately accessible for δ'. In $\delta' \vee \delta''$, neither is δ' immediately accessible to δ'' nor is δ'' immediately accessible to δ'. Accessibility is the transitive closure of immediate accessibility: δ is **accessible** for δ' if there exist γ_i, $1 \leq i \leq n$, such that $\gamma_1 = \delta'$, $\gamma_n = \delta$, and for all $i < n$ the DRS γ_i is immediately accessible for γ_{i+1}. An occurrence of a variable in the body of δ is **bound** if there exists a DRS γ accessible for δ whose head contains x. An unbound occurrence is called **free**.

The constructors $\neg$, :, and $\vee$ correspond to negation, implication and disjunction. The first operation corresponds to the standard merge of the DRS. We will call it the **union**, since we will define a different merge on DRSs.

Definition 2.3 *Let $\delta_1 = [V_1 : \Delta_1]$ and $\delta_2 = [V_2 : \Delta_2]$ be two DRSs. The* ***union*** *of δ_1 and δ_2 is denoted by $\delta_1 \cup \delta_2$ and defined by*

$$\delta_1 \cup \delta_2 := [V_1 \cup V_2 : \Delta_1 \cup \Delta_2]$$

Let us show briefly how in DRS we can calculate the meaning of a simple phrase. Let us take the sentence

(2.8) `A big man sees a small rose.`

The intended translation is the following DRS (modulo renaming of variables).

(2.9)

x	y
man′(x)	big′(x)
rose′(y)	small′(y)
see′(x, y)	

For the DRS is true in a model iff there is an a and a b such that a is a big man, b a small rose and a sees b. We assume that nouns and adjectives are given the same interpretation. For example, man is translated by

(2.10)

∅
man′(x)

The indefinite article is translated by

(2.11)

x
∅

And, finally, the verb is translated by

(2.12)

∅
see′(x, y)

We first choose a constituent analysis.

(2.13) ((`A` (`big man`)) (`sees` (`a` (`small rose`))))

When two parts of speech form a constituent, we form union of the respective DRSs. Obviously, this will only result in a correct translation if we decide on the proper variables to be inserted into the DRS. For notice that the expression man can also be translated by

(2.14)

∅
man′(y)

Therefore, what we need is the following structure prior to translation into DRS-language.

(2.15) $((\mathtt{A}_x\ (\mathtt{big}_x\ \mathtt{man}_x))(\mathtt{sees}_{x,y}\ (\mathtt{a}_y\ (\mathtt{small}_y\ \mathtt{rose}_y))))$

The indices shall guide the translation in the following way. If there is a single variable z in the DRS and the corresponding expression has index x, then the variable z in the DRS shall be replaced by x. If there are two variables in the DRS, z and z' and the corresponding expression has the indices x and y, then z and z' are replaced by x and y.[2] The annotated expression $(\mathtt{a}_x\ (\mathtt{tall}_x\ \mathtt{man}_x))$ is therefore translated by

(2.16) $$\begin{array}{|c|}\hline x \\ \hline \varnothing \\ \hline\end{array} \cup \left(\begin{array}{|c|}\hline \varnothing \\ \hline \mathsf{tall}'(x) \\ \hline\end{array} \cup \begin{array}{|c|}\hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline\end{array} \right) = \begin{array}{|l|}\hline x \\ \hline \mathsf{tall}'(x) \\ \mathsf{man}'(x) \\ \hline\end{array}$$

Similarly, $(\mathtt{a}_y\ (\mathtt{small}_y\ \mathtt{rose}_y))$ is translated as

(2.17) $$\begin{array}{|l|}\hline y \\ \hline \mathsf{small}'(y) \\ \mathsf{rose}'(y) \\ \hline\end{array}$$

The reader is asked to check that we get the desired translation as the result of

(2.18) $$\begin{array}{|l|}\hline x \\ \hline \mathsf{tall}'(x) \\ \mathsf{man}'(x) \\ \hline\end{array} \cup \left(\begin{array}{|l|}\hline \varnothing \\ \hline \mathsf{sees}'(x,y) \\ \hline\end{array} \cup \begin{array}{|l|}\hline y \\ \hline \mathsf{small}'(y) \\ \mathsf{rose}'(y) \\ \hline\end{array} \right)$$

This algorithm has several drawbacks. First, most of the variable management that the λ-calculus was doing for us previously now has to be done 'by hand'. This is unsatisfactory. In Montague's calculus, we would have to choose only a constituent structure and then a correct translation will be returned. However, as we have seen earlier, even this is too much to be assumed. So, we would ideally like to assume no constituent structure at all. We want a calculus that just takes a string and returns a translation. For that, some of the information concerning the structure must be put into the semantics; this is what we will do starting with the next section. Furthermore, we need to reflect a little bit on the nature of the

[2]Notice that in order to be able to tell which is replaced by which we would have to assume that the head of a DRS is not a set but a sequence.

operation with which we translated the constituent juncture. We have hitherto assumed that it is the union. However, there are good arguments to show that the union is not a good choice.

We call an operation $\bullet$ a *merge* only if it has the following property.

(2.19) $\langle\mathfrak{M}, f\rangle \models \delta \bullet \delta' \quad \Leftrightarrow \quad \langle\mathfrak{M}, f\rangle \models \delta$ and $\langle\mathfrak{M}, f\rangle \models \delta'$

This means that $\delta \bullet \delta'$ is the true conjunction of the two DRSs δ and δ'. For we intend each of the DRSs to supply information about their respective variables. However, it is easy to see that the union fails to have this property. Namely, let α and β be unary predicates and $\mathfrak{M} := \langle\{a, b\}, I\rangle$ with $I(\alpha) := \{a\}$, $I(\beta) := \{b\}$. Let f be any assignment. Then

$$\langle\mathfrak{M}, f\rangle \models [x : \alpha(x)]; [x : \beta(x)]$$

(Note that we do not write the head and body in the usual set notation. Typically, we just write the items separated only by a comma; that is to say, we drop the set braces.) However, we do not have

$$\langle\mathfrak{M}, f\rangle \models [x : \alpha(x), \beta(x)]$$

A somewhat simpler example is $\delta := [x : \varnothing]$ and $\delta' := [\varnothing : \alpha(x)]$ and f any function such that $f : x \mapsto b$. Then $\langle\mathfrak{M}, f\rangle \models [x : \alpha(x)]; [x : \varnothing]$ but $\langle\mathfrak{M}, f\rangle \not\models [\varnothing : \alpha(x)]$. However, in this example we have a DRS which has a variable in the body that is unbound. In the sequel we will exclude this case. So, the union is not a good merge. The problem is that we take the union of the heads rather than the disjoint union. Note namely that an occurring of the variable x in the head of δ means *there is an x such that* δ and that likewise an x in the head of δ' means *there is an x such that* δ'. It surely does not follow that *there is an x such that* δ *and* δ', because it might happen that the x satisfying δ are different from the x satisfying δ'. This is why we have to separate the sets of variables of δ and δ'. This we do as follows. Variables get superscripts consisting of sequences of 1s and 2s. These superscripts can be arbitrarily long but finite. Now, for a set V of variables we write

(2.20) $V^1 := \{x^1 : x \in V\}$

So, if $x = v^{\alpha} \in V$ then $x^1 := v^{\alpha 1} \in V^1$. For a formula ϕ let ϕ^1 the result of substituting for every occurrence of a variable x the variable x^1, for every variable x. Likewise for a DRSs $\delta = [V : \Delta]$, let $\delta^1 := [V^1 : \Delta^1]$, where Δ^1 is the the result of replacing each DRS $\gamma \in \Delta$ by γ^1.

Definition 2.4 *Let $\delta = [V : \Gamma]$ and $\delta' = [W : \Delta]$ be two DRSs such that no variable occurs free. Then the* **merge** *of δ with δ', is defined by*

$$\delta \bullet \delta' := [V^1 \cup W^2 : \Gamma^1 \cup \Delta^2]$$

The reader may check that

(2.21) $\delta \bullet \eta = \delta^1 \cup \eta^2$

We shall show that the just defined operation indeed is a merge. To that end, assume that

(2.22) $\langle \mathfrak{M}, f \rangle \models [V^1 \cup W^2 : \Gamma^1 \cup \Delta^2]$

Put $X := V^1 \cup W^2$. Then there exists a $g \sim_X f$ such that $\langle \mathfrak{M}, g \rangle \models \gamma$ for all $\gamma \in \Gamma^1 \cup \Delta^2$. Put $h_1(x) := g(x^1)$ for all $x \in V$, and $h_1(x) := f(x)$ otherwise. Likewise put $h_2(x) := g(x^2)$ for all $x \in W$ and $h_2(x) := f(x)$ otherwise. Now $h_1 \sim_V f$ and $h_2 \sim_W f$. It is an easy matter to verify that for every $\gamma \in \Gamma$

(2.23) $\langle \mathfrak{M}, g \rangle \models \gamma^1 \quad \Leftrightarrow \quad \langle \mathfrak{M}, h_1 \rangle \models \gamma$

and that for every $\delta \in \Delta$

(2.24) $\langle \mathfrak{M}, g \rangle \models \delta^2 \quad \Leftrightarrow \quad \langle \mathfrak{M}, h_2 \rangle \models \delta$

Hence, $\langle \mathfrak{M}, f \rangle \models [V : \Gamma]$ and $\langle \mathfrak{M}, f \rangle \models [W : \Delta]$. Conversely, let $\langle \mathfrak{M}, f \rangle \models [V : \Gamma]; [W : \Delta]$. Then $\langle \mathfrak{M}, f \rangle \models [V^1 : \Gamma^1]$ as well as $\langle \mathfrak{M}, f \rangle \models [W^2 : \Delta^2]$. (Here we need that every variable is bound.) So there exists an h^1 such that $\langle \mathfrak{M}, h^1 \rangle \models [V^1 : \Gamma^1]$ and an h^2 such that $\langle \mathfrak{M}, h^2 \rangle \models [W^2 : \Delta^2]$. Since V^1 and W^2 are disjoint, the following is well-defined: $g(x) := h^1(x)$ if $x \in V^1$, $g(x) := h^2(x)$ if $x \in W^2$ and $g(x) := f(x)$ else. Then $\langle \mathfrak{M}, g \rangle \models \Gamma^1$ and $\langle \mathfrak{M}, g \rangle \models \Delta^2$ and so $\langle \mathfrak{M}, g \rangle \models \Gamma^1 \cup \Delta^2$. Therefore, $\langle \mathfrak{M}, f \rangle \models [V^1 \cup W^2 : \Gamma^1 \cup \Delta^2]$.

Let us finally return to unbound variables. In a DRS $[\varnothing : \alpha(x)]$ the variable x occurs free. Likewise in $[\varnothing : \beta(x)]$. In this case, we do have

(2.25) $$\langle \mathfrak{M}, f \rangle \models [\varnothing : \alpha(x) \cup \beta(x)] \quad \Leftrightarrow \quad \begin{array}{r} \langle \mathfrak{M}, f \rangle \models [\varnothing : \alpha(x)] \\ \textit{and} \quad \langle \mathfrak{M}, f \rangle \models [\varnothing : \beta(x)] \end{array}$$

Hence, free occurrences should in fact not be renamed. This will make the definition of the proper merge rather cumbersome, and we have therefore excluded that case. Notice that also that in our translation we cannot define the union simply by the merge as just defined, since we made crucial use of free variables. Rather, the whole machinery has to be changed. First of all, we do not allow any free variables. Therefore, `man` and `see` are translated by

(2.26)
$$\begin{array}{|l|}\hline x \\ \hline \mathsf{man}'(x) \\ \hline \end{array} \qquad \begin{array}{|l|}\hline x \quad y \\ \hline \mathsf{see}'(x,y) \\ \hline \end{array}$$

By DRT interpretation, these translations mean *there is a man* and *something sees something*. Hence, the indefinite article has lost its function. This is not so tragic. Indeed, many languages do not even have an indefinite article; moreover, it is still not without function for syntactically it is often needed as a left boundary marker for a noun phrase. If we now translate the sentence using the merge, we would of course get a horribly wrong translation, which can be paraphrased as follows: *there is something, there is a big thing, there is a man, something sees something, ...*. We now have the opposite problem: variables are distinct when they should be equal. As we shall see, this is a much more favourable position to be in. What we will now try to achieve is the following: we will assume that the words in addition to the DRSs also contain some information as for how the variables should be handled when the DRS is merged with another one; in particular, we need information as for which variables should in fact be the same after merge. So, by some means two DRSs that are merged should be able to communicate with each other the idea that certain of their variables are actually talking about the same individual. Exactly this information is hidden in the syntax and should be brought to light. This leads us directly to the next section.

2.3 A New Theory of Semantic Composition

In [96] and [95], Kees Vermeulen and Albert Visser have formulated a new theory of meaning. Its philosophy is that the mechanism for gluing meanings is not function application but a rather articulated semantic merging operation. The primary reason for them to assume this is that they want to create an interpretation mechanism that satisfies several conditions. First, any part carries meaning, and gluing certain parts together is basically the same as heaping up meanings. So, rather than determining the meaning by applying a function to an argument we simply take the conjunction of such meanings. This is reasonable because in many cases it is impossible to say which of the two items is a function and which one is the argument. Adjectives and adverbs are a case in point. Second, interpretation works strictly left to right, is fully associative, and allows for starting at any point in the discourse. The latter property is called the *break-in-principle*. This is motivated by the fact that discourse is linear, and the constituent structure which we use in Montague Semantics to assemble the meaning of a sentence has to be derived from the string. The information concerning the sentence structure is encoded into the linear order and the morphology of the words. The latter is very important for our purposes. We wish to bring to light exactly those parts of speech that are concerned with the composition of meaning.

In addition, as we have observed earlier, alternative formalisms such as DRT and Dynamic Montague Grammar [40] have the problem that the choice of variables becomes a precondition on the felicitousness of the interpretation at points where it shouldn't. When inserting the meaning of an item into a structure—say $\mathsf{man}'(x)$—the choice of the variable should be immaterial, because any other variable is just as fine for the mere meaning of that item. But at the point of insertion there might be an accidental capture of that variable, and this has to be prevented. In Montague's own system this does not arise in this particular form since we do not allow free variables. However, as soon as binding facts are to be accounted for, a notion of identity of bound variables is to be reintroduced, giving rise to the infamous rules of *quantifying in*. Now rather than stipulating this, Vermeulen and Visser let the merging operation itself take care of the variable management. Thus, while Montague would let the machinery of λ-calculus do the variable handling, here it is the semantic system itself that does it. Moreover, in some sense this is the only thing it is doing. The interesting thing here is the way in which the merge operates. If two chunks of meaning m_1 and m_2 are merged into $m_1 \bullet m_2$ (think of m_1 and m_2 as being ordinary formulae, or DRSs) then the merge will make all variables of m_2 distinct from those of m_1 before putting them into a single structure. This is the default case; if however m_1 and m_2 contain information to the effect that a variable is intended to be shared between them, then the merge will refrain from renaming that variable in m_2. Of course, the immediate question is how m_1 and m_2 can make it clear that a variable is to be shared. The solution is quite simple: we introduce a vocabulary by which DRSs can communicate about the status of their variables, whether some of them should be identified and others not. This vocabulary will initially be rather simple but later on it becomes more and more involved.

Definition 2.5 *Let N be a set. A* ***referent system over*** *N is a triple* $\langle I, R, E\rangle$*, where R is a finite set, called the* ***set of referents****, I a partial injective function from N to R, called the* ***import function*** *and E a partial injective function from R to N. N is called the set of* ***names****. If* $I(A) = x$*, x is said to have* ***import name*** *A; and if* $E(x) = A$ *then x is said to have* ***export name*** *A.*

Definition 2.6 *Let N be a set. An* ***N-system*** *over N is a pair* $[\mathfrak{R} : \Gamma]$*, where* $\mathfrak{R} = \langle I, R, E\rangle$ *is a referent system over N and* Γ *a DRS over R.* $\mathfrak{R}$ *is called the* ***argument structure*** *of the N-system and* Γ *the* ***body****.*

Actually, it is also possible to define DRS-like structures by allowing Γ to be a set of formulae and N-systems, respectively. We will not make much use of these extended structures. Moreover, we will have to provide means of handling argument structures inside Γ. That case is therefore put aside here. (However, see Section 3.5.) N-systems are written vertically rather than horizontally. Since in a DRS the variable set is separated from the body by a horizontal line, we use a double line to separate the referent system from the DRS. Further, in order to denote the pair $\langle\vec{x}, [\mathfrak{R} : \Gamma]\rangle$, where $\vec{x}$ is a string of our language with denotation $[\mathfrak{R} : \Gamma]$, we usually put the string on top of the N-system. Such strings are typically quoted by means of rightward slashes (/ . . . /).[3] The following example is for illustration.

	/man/
$\mathfrak{R}$	$\langle$NOM, x, NOM$\rangle$
V	$\varnothing$
Δ	man$'(x)$

The merge of two N-systems is defined in two stages. First, we show how referent systems are merged; the merge of N-systems is then rather straightforward. For the definition of the merge recall the merge of two DRSs. There we used the superscript notation. Here we will make this somewhat more precise. Notice first of all that Vermeulen ([95]) uses the notion of a **referent**, which is distinct from a variable, hence the name **referent systems**. In what is to follow, the terms 'referent' and the 'variable' are synomous. Referents can be identified with addresses of a memory cell. The particular address is unimportant as long as we can properly manage these addresses. (Think of the choice of variable names in Prolog.)

[3]These slashes quote morphological sequences. If we want to quote the actual phonological/graphic string we use [. . .].

Referents are featureless objects, they can be distinct or equal; nothing more is important. Our referents have the form $v(\sigma)$, where v is a variable from our set of variables, and σ is an element of $\{1, 2\}^*$, that is, a finite sequence of 1s and 2s. We use x, y and z as metavariables for referents. If $x = v(\sigma)$, then $x^1 := v(\sigma^\frown 1)$ and $x^2 := v(\sigma^\frown 2)$. Using these sequences is a good way to track occurrences of a variable. Now for the definition of the merge. We present some examples first. We write $[A : x : B]$ or simply $A : x : B$ to say that x is imported under the name A and exported under the name B. (So, $I(A) = x$ and $E(x) = B$.) If x has no import name we write $- : x : B$, if it has no export name we write $A : x : -$; we write $- : x : -$ if x has neither an import name nor an export name. A referent system is simply a set of triples $[\alpha : x : \beta]$ where $x \in R$ and $\alpha, \beta \in N \cup \{-\}$. Suppose we merge two referent systems $\{A : x : B\}$ and $\{C : y : D\}$. Then several cases may arise. (1) All four names are distinct. Then x and y are made distinct by using a superscript 1 and 2, and the resulting referent system is $\{A : x^1 : B, C : y^2 : D\}$. (2) $B = C$, and all other names are distinct. Then x and y are taken to be the same variable, which is x^1 (to make the definitions uniform). The resulting referent system is $A : x^1 : D$. Several more cases need to be distinguished, depending on whether A, B, C or D are absent. The easiest cases are A and D. If the first system is $- : x : B$ we get $\{- : x^1 : B, C : y^2 : D\}$ in the first case and $- : x^1 : D$ in the second case. If D is absent. we get $\{A : x^1 : B, C : y^2 : -\}$ in the first case and $A : x^1 : -$ in the second. If B is absent, we get $\{A : x^1 : -, C : y^2 : D\}$, and if C is absent we get $\{A : x^1 : B, - : y^2 : D\}$.

Now there are other possibilities still. We also have to take care of referents that compete for the same import name and referents that compete for the same export name. The first situation arises when $A = C$ the second if $B = D$ (while the other names are assumed different). In the first case, the second referent loses its name, and the first one keeps it, and in the second case it is the other way around. Thus we have $\{A : x : B\} \bullet \{A : y : C\} = \{A : x^1 : B, - : y^2 : C\}$ and in the second case $\{A : x : B\} \bullet \{C : y : B\} = \{A : x^1 : -, C : y^2 : B\}$.

If in the merge x has the export name that y imports, we say that x *supervenes* y. If x and y compete for the same import name, x *I-preempts* y, and if they compete for the same export name, y *E-preempts* x. These situations can arise in all combinations.

Definition 2.7 *Let* $\rho_1 = \langle I_1, R_1, E_1 \rangle$ *and* $\rho_2 = \langle I_2, R_2, E_2 \rangle$ *be referent systems over* N. *Let* $x \in R_1$ *and* $y \in R_2$. *We say that* x^1 ***supervenes*** y^2 *if* $I_2(E_1(x)) = y$. *We say* *that* x^1 ***I-preempts*** y^2 *if there is a* $A \in N$ *such that* $I_1(A) = x$ *and* $I_2(A) = y$*. We say that* y^2 ***E-preempts*** x^1 *if* $E_1(x) = E_2(y)$*.*

Given $\rho_1 = \langle I_1, R_1, E_1 \rangle$ and $\rho_2 = \langle I_2, R_2, E_2 \rangle$ then $\rho_3 := \rho_1 \bullet \rho_2$ is formed as follows. First R_3 is defined. Let $R_1^1 := \{x^1 : x \in R_1\}$ and $R_2^2 := \{x^2 : x \in R_2\}$. Then let $S := \{y \in R_2 : (\exists x \in R_1)(I_1(y) = E_2(x))\}$ be the set of supervened referents and $R_3 := (R_1^1 \cup R_2^2) - S$. This construction ensures that the sum of the sets is disjoint. Next, we define two injections, $\iota_1 : R_1 \rightarrowtail R_3$ and $\iota_2 : R_2 \rightarrowtail R_3$, by

$$
\begin{aligned}
\iota_1(x) &:= x^1 \\
\iota_2(x) &:= \begin{cases} y^1 & \text{if } y^1 \text{ supervenes } x^2 \\ x^2 & \text{if } x^2 \text{ is not supervened} \end{cases}
\end{aligned}
\tag{2.27}
$$

The functions I_3 and E_3 are defined as follows (here $f(x) = \uparrow$ means that f is undefined on x and $f(x) = \downarrow$ that f is defined on x).

$$
\begin{aligned}
I_3(A) &:= \begin{cases} I_1(A) & \text{if } I_1(A) = \downarrow \\ I_2(A) & \text{if } I_1(A) = \uparrow \text{ and } I_2(A) = \downarrow \\ \uparrow & \text{else} \end{cases} \\
E_3(u) &:= \begin{cases} E_2 \circ I_2 \circ E_1(x) & \text{if } u = x^1 \text{ and } E_2 \circ I_1 \circ E_1(x) = \downarrow \\ E_2(x) & \text{if } u = x^2 \text{ and } E_2(x) = \downarrow \\ E_1(x) & \text{if } u = x^1, E_1(x) = \downarrow \text{ and } x^1 \\ & \qquad \text{is not E-preempted} \\ \uparrow & \text{else} \end{cases}
\end{aligned}
\tag{2.28}
$$

Definition 2.8 *Let* $\nu_1 = [\rho_1 : \Gamma_1]$ *and* $\nu_2 = [\rho_2 : \Gamma_2]$ *be two N-systems. The merge is defined as follows*

$$\nu_1 \bullet \nu_2 := [\rho_1 \bullet \rho_2 : \iota_1[\Gamma_1] \cup \iota_2[\Gamma_2]]$$

Here, $\iota_j[\Gamma_j]$ *is the result of replacing every referent r occurring in a formula* ϕ *of* Γ_j *by the referent* $\iota_j(r)$*.*

Let us now show how the N-systems solve our previous problem. We take again our sentence

(2.29) `A tall man sees a small rose.`

Figure 2.1: Merge with nonidentical names

$$\begin{array}{|l|}\hline A : x : B \\ \hline x \\ \hline \phi(x) \\ \hline \end{array} \bullet \begin{array}{|l|}\hline C : x : D \\ \hline \varnothing \\ \hline \psi(x) \\ \hline \end{array} = \begin{array}{|l|}\hline A : x^1 : B \quad C : x^2 : D \\ \hline x^1 \\ \hline \phi(x^1) \\ \psi(x^2) \\ \hline \end{array}$$

Figure 2.2: Merge with identical names

$$\begin{array}{|l|}\hline A : x : B \\ \hline x \\ \hline \phi(x) \\ \hline \end{array} \bullet \begin{array}{|l|}\hline B : x : C \\ \hline \varnothing \\ \hline \psi(x) \\ \hline \end{array} = \begin{array}{|l|}\hline A : x^1 : C \\ \hline \phi(x^1) \\ \psi(x^1) \\ \hline \end{array}$$

To get the desired translation we simply assume that there is exactly one name, so $N = \{\star\}$. Furthermore, determiners, adjectives and nouns get interpreted the same way:

(2.30)
$$\begin{array}{c}\texttt{/man/} \\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}\end{array} \quad \begin{array}{c}\texttt{/tall/} \\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \varnothing \\ \hline \mathsf{tall}'(x) \\ \hline \end{array}\end{array} \quad \begin{array}{c}\texttt{/a/} \\ \begin{array}{|l|}\hline \star : x : \star \\ \hline x \\ \hline \varnothing \\ \hline \end{array}\end{array}$$

The verb however has a more interesting N-system.

(2.31)
$$\begin{array}{c}\texttt{/sees/} \\ \begin{array}{|l|}\hline \star : e : \star, \\ \star : x : -, - : y : \star \\ \hline e \\ \hline \mathsf{see}'(e); \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}'(e) \doteq y. \\ \hline \end{array}\end{array}$$

(This is a referent system even though the name $\star$ is used to identify two referents. Notice, namely, that $I(\star) = x$ and $E(x) = \uparrow$ as well as $E(y) = \star$. So, both E and I

are partial injective functions.) First, let us translate `a tall man`. We get

$$\begin{array}{c}\texttt{/a/}\\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline x \\ \hline \varnothing \\ \hline \end{array}\end{array} \bullet \left(\begin{array}{c}\texttt{/tall/}\\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline \varnothing \\ \hline \mathsf{tall}'(x) \\ \hline \end{array}\end{array} \bullet \begin{array}{c}\texttt{/man/}\\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}\end{array} \right)$$

(2.32)

$$= \begin{array}{c}\texttt{/a tall man/}\\ \begin{array}{|l|}\hline \star : x^1 : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline x^1 \\ \hline \mathsf{tall}'(x^1); \\ \mathsf{man}'(x^1) \\ \hline \end{array}\end{array}$$

Similarly, `a small rose` will receive the translation

(2.33)

$$\begin{array}{c}\texttt{/a small rose/}\\ \begin{array}{|l|}\hline \star : x^1 : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline x^1 \\ \hline \mathsf{small}'(x^1); \\ \mathsf{rose}'(x^1) \\ \hline \end{array}\end{array}$$

(We will replace x^1 by x for readability.) Finally, if combine these two with the verb, we get the following result.

$$\begin{array}{c}\texttt{/a tall}\\ \texttt{man/}\\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline x \\ \hline \mathsf{tall}'(x); \\ \mathsf{man}'(x) \\ \hline \end{array}\end{array} \bullet \left(\begin{array}{c}\texttt{/sees/}\\ \begin{array}{|l|}\hline \star : e : \star, \\ \star : x : -, - : y : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline e \\ \hline \mathsf{see}'(e); \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}'(e) \doteq y. \\ \hline \end{array}\end{array} \bullet \begin{array}{c}\texttt{/a small}\\ \texttt{rose/}\\ \begin{array}{|l|}\hline \star : x : \star \\ \hline \end{array}\\ \begin{array}{|l|}\hline x \\ \hline \mathsf{small}'(x); \\ \mathsf{rose}'(x) \\ \hline \end{array}\end{array} \right)$$

(2.34)

$$= \begin{array}{c}\text{/(2.29)/}\\ \begin{array}{|ll|}\hline \star : e^{12} : \star, & \\ \star : x^1 : -, & - : y^{12} : \star \\ \hline \end{array}\\ \begin{array}{|ll|}\hline x^1, \quad y^{12}, \quad e^{12} & \\ \hline \mathsf{tall}'(x^1); & \mathsf{small}'(y^{12}); \\ \mathsf{man}'(x^1); & \mathsf{rose}'(y^{12}); \\ \mathsf{see}'(e^{12}); & \mathsf{act}'(e^{12}) \doteq x^1; \\ \mathsf{thm}'(e^{12}) \doteq y^{12}. & \\ \hline \end{array}\end{array}$$

The reader may check that in this example the merge is fully associative. Therefore, no constituent structure needs to be prescribed beforehand to arrive at the correct translation. This is, as was explained earlier, a welcome feature of the calculus. Nevertheless, it still suffers from various deficiencies. Notice that we have made no use of the names, only of the directionality of the system. So, a simple transitive sentence in an SVO language (or an OVS language, for that matter) will receive a correct translation simply because the verb can distinguish its arguments from the place they occupy with respect to it. The subject is to the left, the object to the right. In all other types, VSO, VOS, OSV and SOV, the verb cannot discriminate its arguments according to the direction. Some other means must be found. One possibility is morphological marking, and this is what we shall propose in a later section. At the moment, however, we shall pick up a rather delicate problem of the argument selection that is still unresolved in the present calculus.

2.4 Ordnung muss sein

The previous section introduced referent systems and N-systems and showed how a rather basic English sentence gets the right translation. We have used referent systems to combining semantics and syntax. The verb has an argument structure which requires the subject to be on the left hand side and the object on the right hand side. The original system has great drawbacks, however. One cannot specify directionality for adjuncts. For adjuncts must pass the name from left to right, and this turns out to be the same as passing it from right to left. Therefore we cannot model the fact that adjectives must be on the left of a noun in English, and that determiners must be on the left of the noun phrase. Additionally, in English the order of the adjective with respect to the noun plays a role in determining the correct interpretation; for if the adjective is on the right it must be interpreted predicatively. Take as an example the following sentences.[4]

(2.35) `He ate the raw meat.`

(2.36) `He ate the meat raw.`

[4] A somewhat clearer case is presented by French. The phrase `un brave homme` 'a brave man' is different from `un homme brave` 'a well-mannered man' (check!). See also the next chapter on adjectives in Georgian which inflect differently depending on whether they are preposed or postposed.

Moreover, relative clauses never appear on the left side, always on the right side. These and other facts, irrespective of whether one prefers an explanation using different constituent structures, are unaccounted for by the present system. The problem is its linear character with respect to the management of the referents. There is no correlate of the syntactic structure. To see this, note the following fact.

Definition 2.9 *Say that in an argument structure x is* ***identified on the left*** *if* $[A : x : -]$ *or* $[A : x : B]$ *is part of the argument structure, and dually that x is* ***identified on the right*** *if the argument contains* $[- : x : B]$ *or* $[A : x : B]$*. If the argument structure contains* $[- : x : -]$*, x is* ***unidentified****.*

Proposition 2.10 *Let α be an argument structure identifying x only to the right (left). (1) Then $\alpha \bullet \beta$ identifies x^1 only on the right (left) iff β identifies x on both sides. Otherwise $\alpha \bullet \beta$ does not identify x. (2) $\beta \bullet \alpha$ identifies x^2 only on the left (right) iff β identifies x on both sides. Otherwise $\beta \bullet \alpha$ does not identify x.*

This theorem is not hard to prove. Let us look at the DP `a tall man`. We know that it has to export its variable to the right. Therefore, by the previous theorem, the argument structure of `man` cannot be assumed to identify its variable only on the left. Furthermore, `tall` must identify the variable on both sides. By contrast, the verb identifies its object only on the right, so `rose` is not allowed to identify its variable only to the right. To solve this problem we may assume that subject DPs identify their variable to the right and object DPs to the left. However, this has the disadvantage to fix the configurational properties of the DPs with respect to *all* functors. So, there may not be an object DP which is to the left of its functor. Finally, adjectives are still allowed to be on either side of the noun, for they can modify subject nouns as well as object nouns. This state of affairs is unsatisfactory. Therefore, we shall propose the following changes to the referent systems. We introduce a vertical axis in addition to the horizontal one. The vertical axis is responsible for tracking the constituent structure while the horizontal axis is tracking the order.

Definition 2.11 *Let α be an argument structure. α* ***transforms*** *x if* $[A : x : B]$ *is part of α and $A \neq B$.*

In general, we distinguish between *lexical* and *functional* argument structures. These are meant to correspond to argument structures of lexical and of functional words, respectively. In the case of a transformation we write $x : A$ rather than $A : x : A$. However, we still have to indicate in which way the variable continues to be exported and imported. The following three types can be distinguished.

(a) $[A : x : A]$

(b) $[A : x : -]$

(c) $[- : x : A]$

These three types will be denoted by arrows. We have two kinds of arrows, $\ogreaterthan$ and $\olessthan$. Furthermore, we use diacritics to indicate in which way the name survives when a constituent is formed.

Definition 2.12 *A **directional** or **horizontal diacritic mark** is an element of $H := \{\olessthan, \ogreaterthan\}$. A **directional** or **horizontal diacritic** is a (possibly empty) set of directional diacritic marks. We abbreviate $\varnothing$ by $\bigcirc$ (to avoid confusion) and $\{\olessthan, \ogreaterthan\}$ by $\obar$. Otherwise $\olessthan$ abbreviates $\{\olessthan\}$ and $\{\ogreaterthan\}$ abbreviates $\{\ogreaterthan\}$. A **constituent** or **vertical diacritic mark** is an element of $V := \{\vartriangle, \triangledown\}$. A **constituent** or **vertical diacritic** is a set of vertical diacritic marks. We abbreviate the vertical diacritics as follows. We write $-$ for $\varnothing$, $\vartriangle$ for $\{\vartriangle\}$, $\triangledown$ for $\{\triangledown\}$ and $\lozenge$ for $\{\vartriangle, \triangledown\}$. A **diacritic** is a pair $\partial = \langle \sharp, \flat \rangle$, where $\sharp$ is a vertical diacritic and $\flat$ a horizontal diacritic. (Usually we write simply $\sharp, \flat$ rather than $\langle \sharp, \flat \rangle$.) An AIS is said to **export** its referent if it has the vertical diacritic $\vartriangle$ or $\lozenge$; it is said to **import** its referent if it has the vertical diacritic $\triangledown$ or $\lozenge$.*

(The careful reader will observe later that the sets of diacritic marks have a different character. While the horizontal marks are to be read disjunctively, the vertical marks are to be read conjunctively. So, $\{\olessthan, \ogreaterthan\}$ means that x can have the property $\olessthan$ or the property $\ogreaterthan$. However, $\lozenge$ means: x is both $\vartriangle$ and $\triangledown$.) Let us explain informally what these diacritics mean. $\ogreaterthan$ means that the referent in question can be identified on the right, $\olessthan$ means that it can be identified on the left. $\bigcirc$ means that is necessarily unidentified and $\obar = \{\ogreaterthan, \olessthan\}$ means that it can be identified on both sides. Incidentally, if x is unidentified, then the name itself becomes redundant. The vertical diacritic $\triangledown$ means that the referent is *consumed* (so the argument structure in question is a functor with respect to that variable) and $\vartriangle$ means that the referent in question is *produced* (so the argument structure is an argument with respect to the referent). Instead of talking about consumption and

Figure 2.3: Merge with vertical diacritics

$$\begin{array}{ll}
\langle x : \vartriangle : A\rangle \bullet \langle x : \triangledown : A\rangle = \langle x : - : A\rangle & \langle x : \triangledown : A\rangle \bullet \langle x : \vartriangle : A\rangle = \langle x : - : A\rangle \\
\langle x : \vartriangle : A\rangle \bullet \langle x : \Diamond : A\rangle = \langle x : \vartriangle : A\rangle & \langle x : \Diamond : A\rangle \bullet \langle x : \vartriangle : A\rangle = \langle x : \vartriangle : A\rangle \\
\langle x : \triangledown : A\rangle \bullet \langle x : \Diamond : A\rangle = \langle x : \triangledown : A\rangle & \langle x : \Diamond : A\rangle \bullet \langle x : \triangledown : A\rangle = \langle x : \triangledown : A\rangle \\
\langle x : \Diamond : A\rangle \bullet \langle x : \Diamond : A\rangle = \langle x : \Diamond : A\rangle &
\end{array}$$

production we may also think about passing the referent **down** ($\triangledown$) or **up** ($\vartriangle$). If we have the diacritic $\{\vartriangle, \triangledown\}$ then the referent is consumed and produced, or to use the other metaphor, it is passed up and down. A more standard terminology is the following.

Definition 2.13 *Let α be an argument structure x be a referent. Then α is an x-**head** if x carries the diacritic $\triangledown$, an x-**argument** if x carries the diacritic $\vartriangle$, and an x-**adjunct** if x has the diacritic $\Diamond$. α is an x-**carrier** if x has the diacritic $-$.*

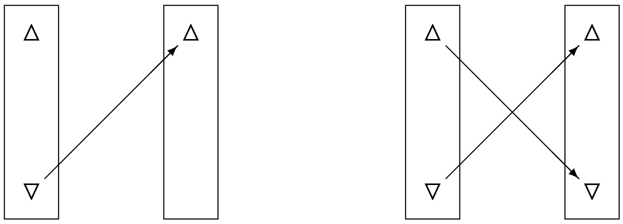
There are several basic scenarios with respect to a referent that is identified in a merge. They can be characterized by their basic behaviour as given in Figure 2.3. Given a specific variable x that is identified under merge, we get the composition table

$$(2.37)\qquad \begin{array}{c|cccc}
 & - & \triangledown & \vartriangle & \Diamond \\ \hline
- & * & * & * & * \\
\triangledown & * & * & - & \triangledown \\
\vartriangle & * & - & * & \vartriangle \\
\Diamond & * & \triangledown & \vartriangle & \Diamond
\end{array}$$

carrier	•	any	=	*
head	•	head	=	*
head	•	argument	=	carrier
head	•	adjunct	=	head
argument	•	adjunct	=	argument
argument	•	argument	=	*
adjunct	•	adjunct	=	adjunct

We think of the vertical diacritics $\vartriangle$ and $\triangledown$ as cancelling each other when the referent is identified. Namely, x is identified exactly when one argument structure

Figure 2.4: Cancellation of Vertical Diacritics



contains x with $\triangle$ and the other x with $\triangledown$. These two diacritics are cancelled, and the remaining diacritics are kept. For example, look at Figure 2.4. Each argument structure can have a head property and an argument property for x. When the argument structures are merged, then one of the two takes up the role of the head and the other that of the argument with respect to x. If the head is to the left, we speak of **rightward merge**; otherwise we speak of **leftward merge**. The respective diacritics are cancelled. (See the figure to the left.) Notice that only in one case it is not already clear in which direction the merge goes: namely in the combination of two x-adjuncts (see the figure to the right). Both can in principle take the role of the head, but in the merge only one is allowed to be the head, while the other must be the argument. This will matter for transformers as well as the horizontal diacritics. Notice that previously we defined transformers to be those argument structures which change the name of the referent. This notion is now applied to the new kind of system in the following way. If a referent has the diacritic $\Diamond$ then instead of a single name we may actually give it a pair $\langle A, B\rangle$ of names, which we write as $A \mapsto B$. We speak of $A \mapsto B$ as a *transformation*. So, we write (single) transformers as $\langle x : \Diamond : A \mapsto B\rangle$. The syntax of transformers is also somewhat tricky. Let us say that in a transformer, A is related to $\triangledown$ and B is related to $\triangle$. So, when x is consumed under the name A it is returned under the name B. The following is now obvious. The merge of two transformers is a transformer, and it is

$$(2.38)\qquad \begin{array}{rcl} \langle x : \Diamond : A \mapsto B\rangle \bullet \langle y : \Diamond : B \mapsto C\rangle & = & \langle x^1 : \Diamond : A \mapsto C\rangle \\ \langle x : \Diamond : A \mapsto B\rangle \bullet \langle y : \Diamond : C \mapsto A\rangle & = & \langle x^1 : \Diamond : C \mapsto B\rangle \end{array}$$

In the first case, the left structure is the argument, since it supplies the name B to be cancelled. Or, to say it differently, the head name C which survives is supplied by the right hand structure. The converse holds for the lower merge. A transformer merged with a nontransformer is a nontransformer. The merge is as follows. Notice however that only one type of combination is possible. The transformer must be the head for the variable, otherwise the merge cannot be defined.

(2.39)
$$\begin{array}{rlcl} & \langle x : \Diamond : A \mapsto B\rangle \bullet \langle y : \vartriangle : A\rangle & = & \langle x^1 : \vartriangle : B\rangle \\ {}^{*} & \langle x : \Diamond : A \mapsto B\rangle \bullet \langle y : \triangledown : X\rangle & = & ? \end{array}$$

Definition 2.14 *Let α and β be argument structures and x a referent. Suppose that x is shared in the merge $\alpha \bullet \beta$. Then α is called a **head (under merge)** relative to x and β an **argument**, if the vertical diacritic of x in α contains $\triangledown$, and if the vertical diacritic of x in β contains $\vartriangle$. If both are heads, we call $\alpha \bullet \beta$ **ambiguous with respect to** x.*

Let us turn now to directionality. The directionality specified on a referent is the direction in which the merge is performed in case that variable is to be shared. The directionality is a property of the head. So, the directional diacritics play a role only if $\triangledown$ is a member of the vertical diacritic. So, we should rather picture the diacritics as follows.

(2.40)
$$\begin{bmatrix} \vartriangle & \\ \triangledown & \olessthan \end{bmatrix}, \quad \begin{bmatrix} \vartriangle & \\ & \bigcirc \end{bmatrix}, \quad \begin{bmatrix} & \\ \triangledown & \obar \end{bmatrix}$$

When a referent is shared, the vertical diacritics are cancelled. The $\triangledown$ of the argument is cancelled together with its directional diacritics, and only the diacritics of the head survive. Hence, the directionality of the shared variable is therefore lost after merge if the complement is an argument. However, the complement may also be an adjunct. In that case it passes on the directionality of its variable. Here we distinguish two cases. (a) If the two directionalities coincide we say that the merge is **harmonic**. (b) If the two directionalities do not coincide, the merge is **disharmonic**. In general, disharmonic merge is not forbidden, though it is a marked option.

(2.41)
$$\begin{array}{clcl} \checkmark & \langle x : \Diamond\ogreaterthan : A\rangle \bullet \langle x : \Diamond\ogreaterthan : A\rangle & = & \langle x : \Diamond\ogreaterthan : A\rangle \\ \checkmark & \langle x : \Diamond\olessthan : A\rangle \bullet \langle x : \Diamond\olessthan : A\rangle & = & \langle x : \Diamond\olessthan : A\rangle \\ \checkmark & \langle x : \Diamond\ogreaterthan : A\rangle \bullet \langle x : \Diamond\olessthan : A\rangle & = & \langle x : \Diamond\olessthan : A\rangle \\ \checkmark & \langle x : \Diamond\ogreaterthan : A\rangle \bullet \langle x : \Diamond\olessthan : A\rangle & = & \langle x : \Diamond\ogreaterthan : A\rangle \\ * & \langle x : \Diamond\olessthan : A\rangle \bullet \langle x : \Diamond\ogreaterthan : A\rangle & = & ? \end{array}$$

Notice that the disharmonic merge succeeds by definition of the merge only when the argument appears on the side where it is identified. That is why the last possibility is excluded. However, when the argument does appear on the side where it is identified, there is an ambiguity as to who is the head and who is the argument. The left argument structure can be the head since it expects its argument on the right side. The result is a left-looking argument structure. On the other hand, the argument structure to the right identifies its argument to the left, so it too can be the head. In that case, however, the resulting argument structure is right-looking.

The case of adjuncts that have no directional requirements (that is, adjuncts which have the diacritic ⦶) is difficult to handle, since it carries both harmonic and disharmonic variants. Since ⦶ stands for the disjunction between the two choices, we will get the results by taking the disjunction over the possibilities. For example, an x-adjunct $\langle x : \lozenge⦶ : A\rangle$ can compose to the right with $\langle x : \lozenge⧁ : A\rangle$ and give $\langle x : \lozenge⧁ : A\rangle$; for x must be harmonic.

Given these restrictions, the following cases remain.

$$\begin{array}{rcl}
\langle x : \lozenge⧁ : A\rangle \bullet \langle x : \vartriangle\bigcirc : A\rangle & = & \langle x : \vartriangle\bigcirc : A\rangle \\
\langle x : \lozenge⧁ : A\rangle \bullet \langle x : \lozenge⧁ : A\rangle & = & \langle x : \lozenge⧁ : A\rangle \\
\langle x : \triangledown⧁ : A\rangle \bullet \langle x : \vartriangle\bigcirc : A\rangle & = & \langle x : -\bigcirc : A\rangle \\
\langle x : \triangledown⧁ : A\rangle \bullet \langle x : \lozenge⧁ : A\rangle & = & \langle x : \triangledown⧁ : A\rangle
\end{array}$$

Let us fix the following notion of a *legal* diacritic. It covers all cases of combinations that are possible and meaningful.

Definition 2.15 *A diacritic ∂ is* **legal** *if (a) $\partial = \langle\{\triangledown\}, \flat\rangle$ or $\partial = \langle\{\vartriangle, \triangledown\}, \flat\rangle$, where $\flat \neq \bigcirc$, or (b) $\partial = \langle -\bigcirc\rangle$ or $\partial = \langle \vartriangle\bigcirc\rangle$. If a diacritic is of the form (b) it is called* ***trivial***.

We remark here that it is actually not necessary to restrict the set of diacritics to the legal ones. For example, we may well allow for a referent with vertical diacritic $\vartriangle$ to specify the order by, say, ⧀. The horizontal diacritic will simply always be ignored by the mechanism. However, the restriction to legal diacritics has the advantage to make the combinatorics of diacritics completely explicit. Let us get back to our English sentence. We will assume the following argument structures and representations for the English nouns, adjectives, determiners and transitive verbs:

$$(2.42)\qquad
\begin{array}{c}\texttt{/man/}\\ \begin{array}{|l|}\hline \langle x : \vartriangle\bigcirc : \bigstar\rangle \\ \hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}\end{array}
\qquad
\begin{array}{c}\texttt{/tall/}\\ \begin{array}{|l|}\hline \langle x : \lozenge\ogreaterthan : \bigstar\rangle \\ \hline \varnothing \\ \hline \mathsf{tall}'(x) \\ \hline \end{array}\end{array}
\qquad
\begin{array}{c}\texttt{/a/}\\ \begin{array}{|l|}\hline \langle x : \lozenge\ogreaterthan : \bigstar\rangle \\ \hline x \\ \hline \varnothing \\ \hline \end{array}\end{array}$$

$$(2.43)\qquad
\begin{array}{c}\texttt{/see/}\\ \begin{array}{|l|}\hline \langle e : \vartriangle\bigcirc : \bigstar\rangle, \\ \langle x : \triangledown\olessthan : \bigstar\rangle, \\ \langle y : \triangledown\ogreaterthan : \bigstar\rangle. \\ \hline e \\ \hline \mathsf{see}'(e); \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}'(e) \doteq y. \\ \hline \end{array}\end{array}$$

The phrase `a tall man` now receives the translation

$$(2.44)\qquad
\begin{array}{c}\texttt{/a/}\\ \begin{array}{|l|}\hline \langle x : \lozenge\ogreaterthan : \bigstar\rangle \\ \hline x \\ \hline \varnothing \\ \hline \end{array}\end{array}
\bullet
\left(
\begin{array}{c}\texttt{/tall/}\\ \begin{array}{|l|}\hline \langle x : \lozenge\ogreaterthan : \bigstar\rangle \\ \hline \varnothing \\ \hline \mathsf{tall}'(x) \\ \hline \end{array}\end{array}
\bullet
\begin{array}{c}\texttt{/man/}\\ \begin{array}{|l|}\hline \langle x : \vartriangle\bigcirc : \bigstar\rangle \\ \hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}\end{array}
\right)$$

$$= \begin{array}{c}\texttt{/a tall man/}\\ \begin{array}{|l|}\hline \langle x^1 : \vartriangle\bigcirc : -\rangle \\ \hline x^1 \\ \hline \mathsf{tall}'(x^1); \\ \mathsf{man}'(x^1). \\ \hline \end{array}\end{array}$$

In this way we capture the following regularities: adjectives precede nouns, determiners precede nouns, subjects precede verbs and objects follow them. However, we cannot derive the fact that determiners must be to the left of the adjectives. This will be implemented later in Section 4.1. Notice that an SVO sentence gets at least two possible parses: S(VO) and (SV)O. This problem will shall address now.

The availability of a SV constituent is a consequence of the lack of order in the argument structure. There is no way to tell a verb in which way it needs to consume its arguments. In addition to the existence of a subject-verb constituent (which one might actually want to have), there are more problems which definitely call for a solution. These are the fact that focus projection in German is may spread to the VP if the object has not been scrambled (an observation due to Tilman Höhle, see [42]). In English we also need to account for order with ditransitive verbs. A ditransitive verb in English must be able to distinguish which of its objects is the first (direct) and which is the second (indirect) object.

(2.45) `They called him an idiot.`

(2.46) `He gave Albert the car.`

Notice namely that inverting the order of the objects results in sentences that are ungrammatical under the same reading as the corresponding (a) sentences.

(2.47) *`They called an idiot him.`

(2.48) *`He gave the car Albert.`

Hence, the verb is forbidden to compose with the indirect object first. However, syntactically, nothing distinguishes an indirect object from a direct object.[5] Hence, if the argument structure is the same we must conclude that the verb can keep track of the order in which the relevant arguments appear. And finally, we do allow *sets of names* rather than names. In fact, when discussing order we have already made use of the possibility to allow disjunction. This will be very essential in explaining the restrictions on word order. Therefore, we will present our final definition.

Definition 2.16 *An* ***argument identification statement*** *or (**AIS**) over N is either (a) a triple* $\langle x : \partial : \mathbb{C}\rangle$*, where x is a referent,* ∂ *a nontrivial legal diacritic, and* $\mathbb{C}$ *a subset of N, or (b) a triple* $\langle x : \partial : \varnothing\rangle$*, where* ∂ *is a trivial legal diacritic or (c) a triple* $\langle x : \Diamond\flat : \mathbb{C}\rangle$*, where* $\flat$ *is a nonempty horizontal diacritic and* $\mathbb{C}$ *a subset of* $N \times N$*. An AIS of type (b) is called* ***empty****. Triples of the form (c) are called* ***transformers****. An* ***argument structure*** *is a sequence* $\alpha = \langle \mu_i : 1 \leq i \leq n\rangle$ *of argument identification statements such that if* $n > 1$ *then* μ_1 *is not empty.*

A few remarks are in order. First, it is technically possible to define the name space of an AIS to be a relation, that is, a subset of $N \times N$, in all three cases. The left hand part of the relation is used for import purposes, and the right hand side for export purposes. This makes some of the definitions more uniform but obscures the nature of the names. Second, we shall use the so-called **attribute**

[5] Not even animacy. It is possible to say `John gave the farmer the slave.` as well as `John gave the slave the farmer.`.

value structures (AVSs) to write down names or sets of names. An AVS is a set of pairs $[A : S]$ where A is an **attribute** and S a set of **values** for A. The set of attributes, P, and the set of values Q are disjoint and fixed beforehand. In an AVS the same attribute can only be used once. However, it need not be used, in which case its set of values is the set Q. We allow for each attribute only a subset $f(A)$ of Q as admissible values. The function f, together with the sets P and Q, determines the space of names. A **name** is an AVS of the form $\{[A : \{x_A\}] : A \in P, x_A \in f(A)\}$. (There are $\prod_{A \in P} |f(A)|$ many names.) If $\mathbb{A}$ and $\mathbb{B}$ are AVSs, we write $\mathbb{A} \leq \mathbb{B}$ if for all attributes A: if $[A : S] \in \mathbb{A}$ and $[A : T] \in \mathbb{B}$ then $S \subseteq T$; and if A does not occur in $\mathbb{B}$ then either it does not occur in $\mathbb{A}$ or $[A : f(A)] \in \mathbb{A}$. It is easy to check that $\mathbb{A} \leq \mathbb{B}$ iff for all names $\mathbb{C}$: if $\mathbb{C} \leq \mathbb{A}$ then $\mathbb{C} \leq \mathbb{B}$.

The notation $\mathbb{A} \mapsto \mathbb{B}$ where $\mathbb{A}$ and $\mathbb{B}$ are AVSs is quite useful but ambigous. This is due to the underspecification that occurs both in $\mathbb{A}$ and in $\mathbb{B}$. We shall assume that attribute value pairs that are not mentioned in $\mathbb{A}$ or $\mathbb{B}$ will not be transformed. Further, we assume that every attribute mentioned in $\mathbb{A}$ is also mentioned in $\mathbb{B}$ and conversely. Finally, the notation $[A : S] \mapsto [A : T]$ will only be used in two cases:

1. Either S or T consist of one element only. In this case $R_A := \{\langle [A : x], [A : y]\rangle : x \in S, y \in T\}$.
2. $[A : S] \mapsto [A : S]$. In this case $R_A := \{\langle [A : x], [A : x]\rangle : x \in S\}$.

The second case obtains for all attributes not mentioned in the AVS, since if A does not occur it effectively stands for the pair $[A : f(A)]$. (The case $f(A) = \{x\}$ is not excluded but not very useful either. If is holds both cases above coincide.) This settles the case of a single feature. If there are several features, the following definition shows how to combine the relations. First, put $R_{\{A\}} := R_A$, and then apply the following definition for S disjoint from T.

(2.49) $$R_{S \cup T} := \{\langle A_1 \cup B_1, A_2 \cup B_2\rangle : \langle A_1, A_2\rangle \in R_S, \langle B_1, B_2\rangle \in R_T\}$$

Thus, we can restrict one feature to certain values, while another feature is set to a particular value, and so on. This eliminates potential confusion.

We shall define equivalence of argument structures as follows.

Definition 2.17 *Two argument structures α and β are* ***equivalent*** *iff β can be obtained from α by adding or removing empty AISs. α is* ***reduced*** *iff it has no empty*

AISs.

The rationale behind this definition is that empty AISs do not contribute to the merge (they might however block merge in certain cases). In fact, we assume that merge of fusion will always result in a reduced argument structure.

Restriction 1 *In a* ***lexical argument structure*** $\langle \mu_i : 1 \leq i \leq n \rangle$*, only* μ_1 *exports its referent. Moreover,* μ_1 *is not a transformer.*

Notice that no requirement is made any more that the exported or the imported names be distinct for distinct referent. In fact, this is quite crucial for the way we handle the merge. The distinctness is anyway not needed since the structure is now ordered. We demand that in a lexical argument structure, it is the first identification statement that carries $\vartriangle$ or $\lozenge$. How can the merge be defined? Clearly, by default, we assume that the order in the sequence matches the order in which the arguments can be taken. For example, the English verb `give` will get (among other) the following semantic structure.

(2.50)

/give/	
$\langle e : \vartriangle\bigcirc : \star \rangle$,	$\langle x : \triangledown\oslash : \star \rangle$,
$\langle y : \triangledown\ominus : \star \rangle$,	$\langle z : \triangledown\ominus : \star \rangle$.
e	
give′(e);	act′$(e) \doteq x$;
thm′$(e) \doteq z$;	goal′$(e) \doteq y$.

We assume that the list of arguments must be processed from right to left and bottom to top. So, the structure takes first the direct object, then the prepositional phrase, and combines with the subject. The directionality assignment is such that the subject ends up on the left side, while the other arguments are on the right side of the verb.

However, we will assume that the merge is somewhat more intricate. First, with respect to the merge with sets of names, let us note that the set is read disjunctively, so the approach we take is a unification approach. If a referent carries a set of names, A_1, A_2, and so on, we assume that it has either name. This means that referents are identified if they share a name in the set (and the diacritics allow sharing). In that case, the common names survive. We now present the complete definition of merge for AISs. Before we do so, some notation must be introduced.

AISs are of the form $\langle x : \Diamond\flat : \mathfrak{A}\rangle$, where $\mathfrak{A} \subseteq N \times N$. Thus $\mathfrak{A}$ is a relation. In all other cases we have $\langle x : \partial : \mathbb{B}\rangle$, with $\mathbb{B} \subseteq N$. Now,

(2.51) $\mathfrak{A} \dashv \mathbb{B} := \{C : \text{ exists } D{:}\langle C, D\rangle \in \mathfrak{A} \text{ and } D \in \mathbb{B}\}$

(2.52) $\mathfrak{A} \vdash \mathbb{B} := \{D : \text{ exists } C{:}\langle C, D\rangle \in \mathfrak{A} \text{ and } C \in \mathbb{B}\}$

(2.53) $\mathfrak{A} \circ \mathfrak{B} := \{\langle C, E\rangle : \text{ exists } E{:}\langle C, E\rangle \in \mathfrak{A} \text{ and } \langle E, D\rangle \in \mathfrak{B}\}$

Definition 2.18 *Let μ and ν be argument identification statements. Rightward merge $\mu \bullet_r \nu$ succeeds in the following cases (only legal diacritics considered).*

(a) $\langle x : \triangledown\ogreaterthan : \mathbb{A}\rangle \bullet_r \langle y : \vartriangle\bigcirc : \mathbb{B}\rangle = \langle x^1 : -\bigcirc : \mathbb{A} \cap \mathbb{B}\rangle$

(b) $\langle x : \triangledown\ogreaterthan : \mathbb{A}\rangle \bullet_r \langle y : \Diamond\olessthan : \mathfrak{C}\rangle = \langle x^1 : \triangledown\ogreaterthan : \mathfrak{C} \dashv \mathbb{A}\rangle$

(c) $\langle x : \triangledown\ogreaterthan : \mathbb{A}\rangle \bullet_r \langle y : \Diamond\olessthan : \mathfrak{C}\rangle = \langle x^1 : \triangledown\ogreaterthan : \mathfrak{C} \dashv \mathbb{A}\rangle$

(d) $\langle x : \Diamond\ogreaterthan : \mathfrak{A}\rangle \bullet_r \langle y : \vartriangle\bigcirc : \mathbb{C}\rangle = \langle x^1 : \vartriangle\bigcirc : \mathfrak{A} \vdash \mathbb{C}\rangle$

(e) $\langle x : \Diamond\ogreaterthan : \mathfrak{A}\rangle \bullet_r \langle y : \Diamond\olessthan : \mathfrak{C}\rangle = \langle x^1 : \Diamond\olessthan : \mathfrak{C} \circ \mathfrak{A}\rangle$

(f) $\langle x : \Diamond\ogreaterthan : \mathfrak{A}\rangle \bullet_r \langle y : \Diamond\ogreaterthan : \mathfrak{C}\rangle = \langle x^1 : \Diamond\ogreaterthan : \mathfrak{C} \circ \mathfrak{A}\rangle$

An entry containing the diacritic $\obar$ is considered to be short for two entries, one containing $\ogreaterthan$ and the other $\olessthan$ place of $\obar$.

Leftward merge $\mu \bullet_\ell \nu$ succeeds in the following cases (only legal diacritics considered).

(a) $\langle x : \vartriangle\bigcirc : \mathbb{A}\rangle \bullet_\ell \langle y : \triangledown\olessthan : \mathbb{B}\rangle = \langle x^1 : -\bigcirc : \mathbb{A} \cap \mathbb{B}\rangle$

(b) $\langle x : \Diamond\olessthan : \mathfrak{A}\rangle \bullet_\ell \langle y : \triangledown\olessthan : \mathbb{C}\rangle = \langle x^1 : \triangledown\olessthan : \mathfrak{A} \dashv \mathbb{C}\rangle$

(c) $\langle x : \Diamond\ogreaterthan : \mathfrak{A}\rangle \bullet_\ell \langle y : \triangledown\olessthan : \mathbb{C}\rangle = \langle x^1 : \triangledown\ogreaterthan : \mathfrak{A} \dashv \mathbb{C}\rangle$

(d) $\langle x : \vartriangle\bigcirc : \mathbb{A}\rangle \bullet_\ell \langle y : \Diamond\olessthan : \mathfrak{C}\rangle = \langle x^1 : \vartriangle\bigcirc : \mathfrak{C} \vdash \mathbb{A}\rangle$

(e) $\langle x : \Diamond\olessthan : \mathfrak{A}\rangle \bullet_\ell \langle y : \Diamond\olessthan : \mathfrak{C}\rangle = \langle x^1 : \Diamond\olessthan : \mathfrak{A} \circ \mathfrak{C}\rangle$

(f) $\langle x : \Diamond\ogreaterthan : \mathfrak{A}\rangle \bullet_r \langle y : \Diamond\olessthan : \mathfrak{C}\rangle = \langle x^1 : \Diamond\ogreaterthan : \mathfrak{A} \circ \mathfrak{C}\rangle$

An entry containing the diacritic $\obar$ is considered to be short for two entries, one containing $\ogreaterthan$ and the other $\olessthan$ place of $\obar$.

As the directionality of the merge is hardly ever free, we shall mostly write just $\bullet$. Also, $\bullet$ will be used both the merge of AISs and for the merge of argument structures. The merge of argument structures is also full of detail, and will be given in the next section.

2.5 Basic Syntax

We shall now turn to the definition of merge and show how basic syntactic facts follow directly from the design of semantic structures. Before we begin, we need to clarify a few things. First, the referent systems are not to be identified with the head section of the DRS. Indeed, technically speaking the head section is nowhere needed in DRT. Instead, we assume that a variable assumes its quantificational force directly from the place where it first occurs. So, we will stop writing a DRS like this: $[x : \mathsf{man}'(x)]$. Rather, we work with the implicit assumption that x occurs in the head section of the highest box that contains it. This is of course not a necessary assumption. It is feasible to assume that our structures are pairs consisting of a referent system, and a genuine DRS, which in turn consists of a head section and a body. Such stacked structures will be necessary to do binding, but for syntactic purposes we can dispense with them. Next we need to see how the merge is defined.

Our basic assumption is that every syntactic merge is accompanied by a semantic merge. So, we assume that whenever two structures $\mathfrak{S}_1$ and $\mathfrak{S}_2$ are merged, so is their meaning. Our syntactic structures are what is called *sign* in the literature. A sign consists in (a) a semantic unit, (b) a syntactic unit and (c) a phonemic unit. However, our analysis will be a little bit different. We will use morphemic representations rather than phonemic representations. This will allow us to concatenate entities that are unpronounceable by themselves (see Chapter 3). Therefore, we will cite these elements using slanted brackets, for example `/house/`. These brackets denote the *morpheme* which is spelled as `house`. We will not analyze the combinatorics of morphemes. For our purposes it will be enough that morphemes can be concatenated. That this will result sometimes in two different words, sometime in one word. That some morphemes change when composed with others, will not bother us here. To give an example, Latin /TANG/ (to touch) when combined with /PERF/ (the perfect morpheme) gives /TANG⌢PERF/, which we also give as /tetig/, because this is the way one would name the perfect stem (loss of nasal, and reduplication). Notice our notational convention here: we write /tang/ to quote the morpheme by its graphemic representation but we write /PERF/ using small caps for an element that has no such representation.

Definition 2.19 *A **sign** is a triple $\langle \vec{x}, \alpha, \Delta \rangle$, where $\vec{x}$ is a string of morphemes, α an argument structure and Δ a DRS such that every unbound referent of Δ occurs in α.*

The merge of two signs is defined as follows. Let $\rho_1 = \langle \alpha_1, \Delta_1 \rangle$ and $\rho_2 = \langle \alpha_2, \Delta_2 \rangle$. Then

(2.54) $$\langle \vec{x}, \rho_1 \rangle \oplus \langle \vec{y}, \rho_2 \rangle := \langle \vec{x}^\frown \vec{y}, \rho_1 \bullet \rho_2 \rangle$$

$\oplus$ is the *merge of representations*. There is an alternative and quite attractive rendering of the facts. It consists in assuming that the merge of argument structures and therefore $\bullet$ is actually commutative. The directionality is only used to determine how the two morpheme sequences are concatenated. So, if the merge identifies x and ρ_1 identifies to its right, then we get $\langle \vec{x}^\frown \vec{y}, \rho_1 \bullet \rho_2 \rangle$, and if it identifies x to its left then we get $\langle \vec{y}^\frown \vec{x}, \rho_1 \bullet \rho_2 \rangle$. We shall not pursue that option here, though.

The definition of merge is split into several cases. First, we shall define the notion of *access*; there are two kinds of access restrictions, exemplified by English and German. Second, we distinguish *merge* from *fusion*; and finally, we distinguish between *monadic* and *polyadic* merge (or fusion). Let us begin with the problem of access to individual AISs within an argument structure. In contrast to the original conception of referent systems, we have argued that argument structure is not a set of AISs but a sequence thereof. In later chapters we shall develop a slightly more articulated view on that matter. The definition of access can be given two forms: either we talk about AISs or we talk about the variable that these AISs contain. We prefer the latter version. Notice however that while in a given argument structure different AISs have different variables, this need not be true across two AISs. For the purpose of the next definitions, we assume that the variables of the first argument structure are x_i and the variables of the second structure y_j.

Definition 2.20 *Let $\alpha = \langle \mu_i : 1 \leq i \leq m\rangle$ and $\beta = \langle \nu_j : 1 \leq j \leq n\rangle$ be argument structures. The rightward merge $\alpha \bullet_r \beta$ is defined iff x_m accesses a variable y_k of β and $\mu_m \bullet_r \nu_k$ is defined. The leftward merge $\alpha \bullet_\ell \beta$ is defined iff y_n accesses a variable x_k of α and $\mu_k \bullet_\ell \nu_n$ is defined.*

Now everything hinges on the notion of access. We assume that access is not uniform across languages. For example, English generally has strict access, while German for example has a more liberal access rule, allowing to jump an AISs if the feature specification does not match.

Definition 2.21 *Let $\alpha = \langle \mu_i : 1 \leq i \leq m\rangle$ and $\beta = \langle \nu_j : 1 \leq j \leq n\rangle$ be argument structures.*

- *x_m **E-accesses** y_j iff $j = n$. y_n **E-accesses** x_i iff $i = m$.*
- *x_m **G-accesses** y_j iff j is maximal such that the merge $\mu_m \bullet \nu_j$ succeeds with x_m being the head. y_n **G-accesses** x_j iff j is maximal such that the merge $\mu_j \bullet \nu_n$ succeeds with y_n being the head.*

The idea behind these definitions is as follows. Suppose that α is the argument structure of a verb looking for the following arguments:

$$(2.55)\qquad \begin{array}{ccccc} e & y_1 & y_2 & y_3 & y_4 \\ & \vdots & \vdots & \vdots & \vdots \\ & \text{NOM} & \text{DAT} & \text{NOM} \sqcup \text{ACC} & \text{ALL} \end{array}$$

The diacritics are as follows. e has $\vartriangle$, all others have $\triangledown$. Let the horizontal diacritics by ⦸ for all y_i, $1 \leq i \leq 4$. Now, let β be the argument structure of an NP. So, it may be depicted by

$$(2.56)\qquad \begin{array}{c} x \\ \vdots \\ \text{CASE} \end{array}$$

Then, with E-access the merge $\alpha \bullet \beta$ will succeed only if CASE matches with ALL. For CASE must match the last entry for β. If that is the case, x E-accesses y_4. If G-access is assumed, the situation is different. If CASE is ALL or matches ALL, then x accesses y_4. If CASE matches NOM or ACC but does not match ALL then x accesses y_3.

If CASE matches DAT but matches neither ALL not ACC nor NOM, then x accesses y_2. And so on. It is clear that if CASE matches NOM then x does not access y_1, but y_3. (In the case of G-access, if names are sets, then it is just required that the intersection is not empty. So the first potential candidate for matching is taken.) Access is defined to be order insensitive. So, the match in name is alone responsible for the recognition of the argument. We feel that this is the right choice, but an order sensitive definition can of course be given.

The syntactic restrictions should therefore fall out of the restrictions on combining argument structures. We will investigate this option here with respect to basic syntax. First, as with argument structures, we take it that there exists a lexical and a functional merge. We assume that lexical elements can only lexically merge, but functional elements have the choice of merging functionally or lexically. First, let us put down the most important of all restrictions.

Restriction 2 *A **merge** of representations can take place only if at least one referent is identified. Moreover, in a **monadic merge** that referent is unique.*

The first condition holds for all types of merge and ensures that only those parts of speech are combined which share some common object about which they speak. The restrictions on lexical argument structure and lexical merge allow us to derive the standard X-bar syntax in the following way.

We distinguish lexical and nonlexical merge. The standard case is the monadic merge. In a monadic merge the following holds:

- For rightward merge: $\alpha = \langle \mu_i : 1 \leq i \leq m \rangle$, and $\beta = \langle \nu_1 \rangle$, where ν_1 does not import y_1; and
- for leftward merge: $\alpha = \langle \mu_1 \rangle$, and $\beta = \langle \nu_i : 1 \leq i \leq n \rangle$, where μ_1 does not import x_1.

If merge is lexical, the direction of merge is also unique. For if β is saturated and y_1 is not a head nor adjunct then it cannot take any x_i as argument. The general (that is to say polyadic) case is defined as follows.

Definition 2.22 *An argument structure is **saturated** if none of its AISs imports any of its referents.*

Restriction 3 *In a **merge** the complement is saturated.*

Let $\alpha = \langle \mu_1, \dots, \mu_m \rangle$ and $\beta = \langle \nu_1, \dots, \nu_n \rangle$ be two argument structures. We define the monadic merge of these argument structures. Let β be saturated. So, α is unsaturated and it is the referent that is passed up by β that is looking for identification. As the identification statements need not have disjoint sets of names, it may well be that several identification statements can identify the referent. In this case, we assume that the rightmost referent wins.

In fusion, both arguments structures may be unsaturated. The availability of fusion has has important consequences. If we only used merge constituents are predicted to be continuous. Consider for example the German adjective `stolz` ('proud'). Suppose it has the translation $\mathsf{proud}'(x, y)$. It is an adjunct with respect to x but has an argument y. If we attempt a merger, then the adjective must be a head. Hence the referent x must be identified first. So we must combine with a phrase of the form `auf seine Schüler` ('of his pupils'). The phrase `auf seine Schüler stolz` is an adjunct, as is the adjective `stolz`. So, if fusion is not an option, we cannot simply merge `stolz` with the head noun `Lehrer` ('teacher') and obtain `stolzer Lehrer` and later merge with `auf seine Schüler`. Thus the following contrast is accounted for:

(2.57) `der auf seine Schüler stolze Lehrer`

(2.58) *`der stolze Lehrer auf seine Schüler`

In contrast to merge, fusion requires a number of additional decision concerning the fate the arguments that the complement brings into the new structure. Suppose that α is fused with β and that α is in need of the arguments γ_1, γ_2, while β needs the argument δ. Then in which order does $\alpha \bullet \beta$ need its arguments? There are three choices that come to mind:

$$(2.59) \qquad \begin{array}{ccc} \delta & \gamma_1 & \gamma_2 \\ \gamma_1 & \delta & \gamma_2 \\ \gamma_1 & \gamma_2 & \delta \end{array}$$

Obviously, we must make a choice here. In Chapter 6.5 we shall look at a particular construction, namely raising verbs, where this is relevant. Our analysis assumes that verbs selecting verbs can actually signal which of the options they wish to realize. If they select one option they become control verbs and if they select the other they become raising verbs.

Finally, we must consider a last possibility: that merge identifies several variables in one step. Such a merge will be called **polyadic merge**.

Definition 2.23 *Let $\alpha = \langle \mu_1, \ldots, \mu_m \rangle$ and $\beta = \langle \nu_1, \ldots, \nu_n \rangle$ be argument structures. Then the polyadic merge $\alpha \bullet \beta$ succeeds iff the followin holds:*

① *in a rightward merge: there are sequences $i_1, \cdots, i_p$ and $j_1, \cdots, j_p$, $p > 0$, such that*

(a) $i_1, \ldots, i_p$ is strictly descending and $j_1, \ldots, j_p$ is strictly ascending;

(b) $\mu_{i_k} \bullet \nu_{j_k}$ is a rightward merge for all $k \leq p$;

(c) all AISs of β that export a referent are in the list of ν_{j_k};

(d) the monadic rightward merge $\alpha \bullet \langle \nu_1 \rangle$ succeeds.

In this case, the resulting argument structure is $\langle \xi_u : i \leq m \rangle$, where $\xi_u := \mu_{i_k} \bullet_r \nu_{j_k}$ if $u = i_k$, and $\xi_u := \alpha_u$ else.

② *in a leftward merge: there are descending sequences $i_1, \cdots, i_p$ and $j_1, \cdots, j_p$, $p > 0$, such that*

(a) $i_1, \ldots, i_p$ is strictly ascending and $j_1, \ldots, j_p$ is strictly descending;

(b) $\mu_{i_k} \bullet \nu_{j_k}$ is a leftward merge for all $k \leq p$;

(c) all AISs of α that exported a referent are in the list of μ_{j_k};

(d) the monadic leftward merge $\langle \mu_1 \rangle \bullet \beta$ succeeds.

In this case, the resulting argument structure is $\langle \xi_u : i \leq m \rangle$, where $\xi_u := \mu_{i_k} \bullet_\ell \nu_{j_k}$ if $u = j_k$, and $\xi_u := \nu_u$ else.

(In this definition, some of the ξ_i will be empty, in which case they will be dropped. However, for the purpose of the definition it is easier to keep them first.) This means that in a polyadic merge, the first member of the sequence must be merged as in a monadic merge. This merge determines the direction of the merge. The other merge operations have to follow the same directionality. They must be orderable in decreasing order. (In fact, we hardly need more than two simultaneous mergers.) Moreover, the exported variables of the argument must all be merged; none may be left out. If we attempt a merge between α and β then the choice of monadic versus polyadic merge is determined solely by the nature of the argument, not the functor. Every referent that the AISs of the argument export must be merged into the functor. One may wonder whether polyadic E-merge should actually be stricter and require that the ascending sequence be uninterrupted. This would be like an iterated E-merge, discharging one pair after the other. In practice, there is no situation where this makes a difference.

Finally, there is also polyadic fusion. Here we shall have to state what to do with the arguments that the functor inherits from its argumemnt. We shall assume that they are appended at the end of the entire structure.

There is a simple four-fold classification of lexical argument structures given two basic objects, events and objects. In a lexical argument structure one variable is exported, and this variable determines the semantic sort of the result. Depending on whether this variable is also imported or not, the *category* of a syntactic object derives from its argument structure as follows.

	$\triangle$	$\Diamond$
event	V	Adv
thing	N	A

Let us explain this a little bit. Nouns export a referent denoting objects. There are nouns that import no other variable, such as `man`, whereas others do import one, such as `destruction` or `father`. Nouns are prototypical arguments with respect to the referent that they export. Adjectives and adverbs are adjuncts with respect to the referent that they export. This explains why they can be accumulated in any number within a noun or verb phrase. They, too, may select arguments, such as `proud`. Verbs have an event referent, which is external, but may take a number of arguments. There are still more categories, for example prepositions. The problem with classifying prepositions is that they may be used to modify events as well as objects. Hence, prepositions may function as adverbs as well as adjectives. This means that the variable they create is not type restricted. Moreover, we will see that prepositions are often not lexical.

Now we turn to X-bar syntax. The basic property of X-bar syntax is that there is a head and it projects up to a phrase. In Government and Binding, the types of syntactic junctures shown in Figure 2.5 are allowed (order irrelevant). Here, X and Y are variables over categories. The primes count the levels of projection. There are zero, one and two primes, hence up to three levels. The variable X

Figure 2.5: X-bar Syntax

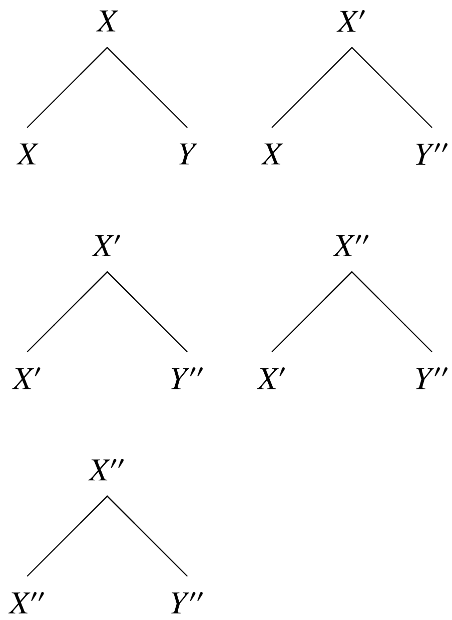
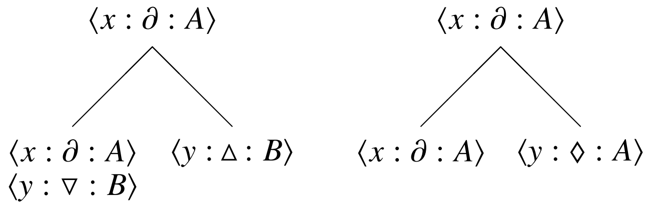


Figure 2.6: Argument Discharge



denotes the head, since it is both the category of a daughter and of the mother.[6] We have already seen how the constraints on lexical argument structure allow to define the category of a word or structure in the usual sense. Let us now see how the conditions on lexical merge allow to deduce the basic properties of X-bar syntax. First, an argument structure is phrasal iff it does not import any referent iff it is saturated. In X-bar terms this means that it is of the third level (two primes). In our terms, however, no levels are assigned to a phrase. Hence, there can be any number of levels in between a word and its corresponding phrase, although that number will rarely exceed three. For it directly corresponds to the number of arguments a word-level argument structure needs. If the highest is three, our highest level will be four. We have two basic types of merge: head-complement and head-adjunct. These are exemplified in Figure 2.6. To the left we have the combination head-complement. One referent, different from the one defining the category, is identified and 'discharged' from the argument structure. The level increases, since there are less argument discharges needed to reach the phrasal level. To the right we have the head-adjunct juncture. The adjunct identifies the referent, but no change is made in the argument structure of the head. Notice that the referent that gets identified is not necessarily the head referent. One has to remember that our calculus allows for junctures that are not X-bar syntax proper. Therefore, no step is taken to ban such junctures. Finally, we have to discuss the head-head juncture in X-bar syntax. This is used differently in Government and Binding. Namely, this is not a phrasal combination but at word level, as we can see from the fact that the level is not increased. Moreover, it covers such cases as serial verbs, the verbal cluster tense+aspect+verb and so on. Hence, it corresponds in our terminology not to the lexical merge but to the functional merge. Since the functional merge is rather involved we will not discuss it here.

---

[6] It is allowed that $X = Y$.

We will exemplify the effect of these assumptions with Latin and German syntax. In Latin, word order is quite free. This pertains foremost to the arguments of the verb. Consider as an example the following sentence.

(2.60) `Cicero consuli librum dat.`
Cicero gives a/the book to the consul.

This sentence is grammatical in all 24 permutations of the words. The uninflected verb has the following argument structure.

(2.61)

`/dat/`

$\langle e : \triangle\bigcirc : - \rangle$,	$\langle x : \nabla\text{⦶} : \textsc{nom} \rangle$,
$\langle y : \nabla\text{⦶} : \textsc{acc} \rangle$,	$\langle z : \nabla\text{⦶} : \textsc{dat} \rangle$
e	
give′(e);	act′$(e) \doteq x$;
thm′$(e) \doteq y$;	ben′$(e) \doteq z$.

Notice that no direction is specified. Then the verb may merge with any argument on either side. So, let us give the following structures to the remaining words:

(2.62)

`/Cicero/`	`/librum/`	`/consuli/`
$\langle x : \triangle\bigcirc : \textsc{nom} \rangle$	$\langle x : \triangle\bigcirc : \textsc{acc} \rangle$	$\langle x : \triangle\bigcirc : \textsc{dat} \rangle$
x	x	x
cicero′(x)	book′(x)	consul′(x)

Then all 24 combinations are acceptable and result in the following structure (modulo renaming of referents):

(2.63)

$\langle e : \triangle\bigcirc : - \rangle$	
$x, \quad y, \quad z, \quad e$	
give′(e);	act′$(e) \doteq x$;
thm′$(e) \doteq y$;	ben′$(e) \doteq z$;
cicero′(x);	book′(y);
consul′(z).	

Notice that this forces the constituent structure to be as follows. (We list 4 of the 24 possibilities, which correspond to the argument order SIO.)

S(I(OV)) S(I(VO)), S((IV)O), (S(IV))O
S((VI)O), (S(VI))O, ((SV)I)O ((VS)I)O

So, some sentences receive three possible constituent structures, namely when the verb is in medial position. For then it has a choice to combine with the argument to the left or with the argument to the right. In this respect, this analysis shares some feature with Combinatory Categorial Grammar (CCG). The latter also allows for coexisting constituent structures. It is worthwhile to compare the present approach with that of Steedman, taken from [90]. We will apply some charity in assuming the nondirectional slash throughout, written $\multimap$. This gives the following type assignment, if we start with the basic types of truth value, *t*, and object, *e*. (For comparison, Steedman's VP is our $\mathsf{e} \multimap \mathsf{t}$, and his NP is our e.)

sentence	:	v_0	:=	t
intransitive verb	:	v_1	:=	$\mathsf{e} \multimap \mathsf{s}$
transitive verb	:	v_2	:=	$\mathsf{e} \multimap \mathsf{v}_1$
ditransitive verb	:	v_3	:=	$\mathsf{e} \multimap \mathsf{v}_2$
subject	:	su	:=	$\mathsf{v}_1 \multimap \mathsf{v}_0$
indirect object	:	io	:=	$\mathsf{v}_2 \multimap \mathsf{v}_1$
direct object	:	do	:=	$\mathsf{v}_3 \multimap \mathsf{v}_2$

According to Steedman, the function of case is to type raise the noun phrase from type *e* into either subject (nominative), object (accusative) or indirect object (dative). We ignore the weakness of this approach with respect to idiosyncratic case assignment. More grave is following problem. Consider again the sentence above. The following two variants are ungrammatical in CCG.

(2.64) `Consuli dat Cicero librum.`

(2.65) `Librum Cicero dat consuli.`

Since these are mirror images of each other, lets only discuss the first. It corresponds to the order IVSO. Let us resolve the category symbols one step. Then we get

(2.66) $\mathsf{v}_2 \multimap \mathsf{v}_1 \quad \mathsf{v}_3 \quad \mathsf{v}_1 \multimap \mathsf{v}_0 \quad \mathsf{v}_3 \multimap \mathsf{v}_2$

On this level, no two adjacent categories can be combined. Let us therefore unravel the abbreviations completely.

(2.67) $(\mathsf{e} \multimap (\mathsf{e} \multimap \mathsf{t})) \multimap (\mathsf{e} \multimap \mathsf{t}) \quad \mathsf{e} \multimap (\mathsf{e} \multimap (\mathsf{e} \multimap \mathsf{t})) \quad (\mathsf{e} \multimap \mathsf{t}) \multimap \mathsf{t}$
$(\mathsf{e} \multimap (\mathsf{e} \multimap (\mathsf{e} \multimap \mathsf{t}))) \multimap (\mathsf{e} \multimap (\mathsf{e} \multimap \mathsf{t}))$

We are allowed to compose $\alpha \multimap \beta$ with any category that has an argument of the form β, and replace that argument by α. So, S and O may combine to

$$(2.68) \qquad (\mathsf{e} \multimap (\mathsf{e} \multimap (\mathsf{e} \multimap \mathsf{t}))) \multimap (\mathsf{e} \multimap \mathsf{t}) = \mathsf{v}_3 \multimap \mathsf{v}_1$$

which corresponds to (NP $\multimap$ NP $\multimap$ VP)) $\multimap$ VP in the nondirectional version of CCG. This can compose with the verb to give a category v_1. This cannot compose with the indirect object. The reader may check that there is only one more possibility, namely that the verb and the subject compose to $\mathsf{v}_2 \multimap \mathsf{v}_0$. This can compose with the object to $\mathsf{v}_3 \multimap \mathsf{v}_0$, after which no more rules can be applied. So, these two word orders cannot be accounted for in Steedman's system. On the other hand, Steedman's main concern is to get the gapping properties of languages right. The facts are argued to be as follows (following [83] and [64]).

	VSO:	VSO and SO	*SO and VSO
(2.69)	SOV:	*SOV and SO	SO and SOV
	SVO:	SVO and SO	*SO and SVO

There is an additional complication, namely that the purported fact of SOV languages—namely, that sentences of the form 'SOV and SO' are ungrammatical—does not seem to be correct (this is acknowledged in Steedman's article). In German, such sentences are good, as shows the following

(2.70) `Ich sagte, daß Josef Klavier spielt und Paul Gitarre.`
I said that Josef Piano plays and Paul guitar.

So, the directionality facts do not fall out as nicely as claimed. Moreover, as constituents must be connected strings, gapping in verb medial languages is predicted to be severely restricted. For example, the following English sentence would in this approach be ungrammatical

(2.71) `I play a sonata and you a concerto.`

In order to circumvent this (and to get at the German facts), Steedman introduces a reanalysis rule that allows to disconnect in retrospect the sentence into SO and V, which are then constituents and gapping is licensed. Whatever this may be good for, we are more inclined to assume a theory of gapping that uses deletion, perhaps as in Wilder [99], [100]. We will not pursue this theme further.[7] Let us

[7] Also the theory of Wilder is based on some questionable assumption such as the big conjunct

now turn to German. German is like Latin. However, the verb must be to the right of the argument. Hence the semantic structure for the verb `geben` (*give*) is

(2.72)

/geben/	
$\langle e : \triangle\bigcirc : -\rangle$,	$\langle x : \nabla\oslash : \text{NOM}\rangle$,
$\langle y : \nabla\oslash : \text{ACC}\rangle$,	$\langle z : \nabla\oslash : \text{DAT}\rangle$.
e	
$\mathsf{give}'(e)$;	$\mathsf{act}'(e) \doteq x$;
$\mathsf{thm}'(e) \doteq z$;	$\mathsf{ben}'(e) \doteq z$.

(This will account only for the word order of subordinate clauses.) Hence the following word orders are legitimate

(2.73) SOIV, SIOV, OSIV, OISV, ISOV, ISOV

This is correct. Moreover, the theory predicts another fact, namely that adverbs (or adverbial phrases for that matter) may occur at any position between the arguments, that is, at any place marked by a star

(2.74) $\star S \star O \star I \star V$

For adverbs have the argument structure $[\langle e : \Diamond\oslash : \alpha\rangle, \ldots]$. In a similar fashion, adjectives appear to the left of the noun and agree with the noun in case, number and gender. Prepositions or postpositions are clear. The basic structure of a subordinate clause is therefore accounted for. Notice that we do not need any scrambling. We will discuss the implications of this later. It is noted here only that under the current assumptions there is not necessarily a unique normal word order, as is assumed by many authors.[8]

Notes on this section. The system defined so far looks quite like categorial grammar. Yet there are noteworthy differences. First, we have defined liberal word orders not by using lexical rules (because the typing system generally is not flexible enough). But even if we were to consider only E-access, our system is

hypothesis, which says that only big categories, can be conjuncts. Wilder subscribes to the view that only DPs and CPs can be conjuncts. This is at odds with the semantics, though, for reasons that we shall not go into. However, it is not necessary to believe in big conjuncts in order to assume that gapping and other phenomena are an effect of deletion rather than reconstruction, as in Steedman's analysis.

[8]Notice that both CCG and this theory do not need the Geach rule. Namely, if adverbs are of the type $\mathsf{v}_0 \multimap \mathsf{v}_0$ then they may compose with v_i and the resulting category is v_i, for every $i \geq 0$.

different. Syntactically, the types that can be defined are more restricted. If order is disregarded, they are of the form

(2.75) $\alpha_1 \multimap (\alpha_2 \multimap \ldots (\alpha_n \multimap \beta) \ldots)$

where the α_i, $1 \leq i \leq n$, and β are basic types. In order for this to work, two things are necessary. First, the basic ontology must be rich enough to accommodate differences that are otherwise accounted for by higher types. Second, the semantics is flexible enough to avoid the need for Geach's rule. For Geach's rule would be needed since one and the same adjunct can combine with different argument structures. Exactly this is the case with the semantics here.

2.6 Syncretism

The formal calculus shows us how particular structures are paired with particular meanings. The operations are simultaneously defined for both. However, in actual communication, we are given only one, form or meaning, and want to obtain the other. That is, we translate from meaning to form when we are the speaker and from form to meaning when we are the hearer. The translation is not unique; the same meaning can expressed in different ways, and the same expression can mean different things. We shall address a particular problem that arises in connection with G-acess: syncretism.

Consider a language that uses the G-access. In such a language, a verb takes its arguments in any order, since it is allowed to take the last *matching element* rather than the last element simpliciter. Hence the verb takes its arguments in any order. This is borne out for German.

(2.76) `..., dass der Kater den Vater sieht.`
..., that the tomcat the father sees.

(2.77) `..., dass den Vater der Kater sieht.`
..., that the father the tomcat sees.
..., that the tomcat sees the father.

(2.78) `..., dass den Kater der Vater sieht.`
..., that the tomcat the father sees.

(2.79) `..., dass der Vater den Kater sieht.`

..., that the father the tomcat sees.

..., that the father sees the tomcat.

Here, (2.76) and (2.77) both mean the same, and (2.78) and (2.79) also mean the same, as shown in the translation. This is precisely as we have seen above in the case of Latin.

Now, German nouns also show a lot of case syncretism. For example there is in general no distinction between nominative, accusative, and `Kater` could also be singular or plural. The burden of disambiguation is entirely on the determiner. In the masculine singular the determiner shows clearly and unambiguously case and number.

We shall now look at cases when the determiner actually fails to exhibit the required contrast. The accusative and the nominative case forms are identical for feminine, neuter and all plural noun phrases. For example, `die Katze` may be either nominative or accusative. The sentence (2.80) is ambigous between two readings, which correspond in English to (2.81) and (2.82).

(2.80) ..., `dass die Katze die Mutter sieht.`

(2.81) *..., that the cat the mother sees.*

(2.82) *..., that the mother the cat sees.*

Let us see how we can account for this possibility.

Let A be the speaker of (2.80). A knows whether he wanted to convey (2.81) or (2.82). He will use the proper case labels in his calculation. If A intends that `die Katze` is the subject, he will give it the case name NOM, and if he assumed that `die Katze` was object, he would give it the case name ACC. Similarly with `die Mutter`. So, for A the situation is as follows for (2.81).

(2.83)
$$\begin{array}{ccccc} \texttt{die Katze} & \texttt{die Mutter} & & \texttt{sieht} & \\ \langle x\rangle & \langle y\rangle & \langle e, & v_1, & v_2\rangle \\ \vdots & \vdots & & \vdots & \vdots \\ \text{NOM} & \text{ACC} & & \text{NOM} & \text{ACC} \end{array}$$

And for (2.82) it is like this:

(2.84)
$$\begin{array}{ccccc} \texttt{die Katze} & \texttt{die Mutter} & & \texttt{sieht} & \\ \langle x\rangle & \langle y\rangle & \langle e, & v_1, & v_2\rangle \\ \vdots & \vdots & & \vdots & \vdots \\ \text{ACC} & \text{NOM} & & \text{NOM} & \text{ACC} \end{array}$$

In both cases, the merge is well-defined with G-access and yields the desired translations.

For the hearer—call him B—the situation is different. B does not know which case labels to stick in. There are now two distinct ways in which he can handle the situation. The first choice is to interpret `die Katze` as the exponent of two (homonynous) phrases, one in the nominative and the other in the accusative. The other is to interpret it as the exponent of a single underspecified structure. These options make different predictions concerning grammatical acceptability which I shall now turn to. Consider the first option. Since the calculus is blind to the actual form of the exponents it turns out to predict that all word orders are grammatical, as if all case endings were maximally distinct.

The second option is different. Under this option B will assume that (2.80) means (2.81). Let us see why this is so. The elements involved are drawn from the lexicon (plus morphology, but see Chapter 3) by looking at their overt form—since this is what we are given. All noun phrases (`die Katze`, `die Mutter`) are ambiguous between nominative and accusative singular. B will therefore represent them as follows:

(2.85)
$$\begin{array}{ccccc}
\texttt{die Katze} & \texttt{die Mutter} & & \texttt{sieht} & \\
\langle x \rangle & \langle y \rangle & \langle e, & v_1, & v_2 \rangle \\
\vdots & \vdots & & \vdots & \vdots \\
\textsc{nom} \sqcup \textsc{acc} & \textsc{nom} \sqcup \textsc{acc} & & \textsc{nom} & \textsc{acc}
\end{array}$$

Here, the merge yields only (2.81) as a result. Let us look at this a little bit closer. The structures to be inserted for the the words are underspecified. For example, we insert the following structures for the words:

(2.86)

/`Katze`/

$\langle x : \triangle \bigcirc :$	NUM : *sg* CASE: *nom, acc, dat*
$\varnothing$	
cat′(x)	

/`Mutter`/

$\langle x : \triangle \bigcirc :$	NUM : *sg* CASE: *nom, acc, dat*
$\varnothing$	
mother′(x)	

Here, we ignore gender for simplicity. (All occurring items are feminine.) The determiner `die` is also in many ways ambiguous:

(2.87)
$$
\begin{array}{c}
\texttt{/die/} \\
\begin{array}{|l|}
\hline
\langle x : \Diamond\ominus : \left[\begin{array}{l}\textsc{num}: sg,\ pl \\ \textsc{case}: nom,\ acc,\ dat\end{array}\right] \rangle \\
\hline
x \\
\hline
\mathsf{unique}'(x) \\
\hline
\end{array}
\end{array}
$$

(The semantics should not be taken too seriously now.) If we merge the structures for `die` and `Katze` and `die` and `Mutter`, respectively, we obtain for the first pair

(2.88)
$$
\begin{array}{c}
\texttt{/die Katze/} \\
\begin{array}{|l|}
\hline
\langle x : \triangle\bigcirc : \left[\begin{array}{l}\textsc{num}: sg \\ \textsc{case}: nom,\ acc\end{array}\right] \rangle \\
\hline
x \\
\hline
\mathsf{cat}'(x); \mathsf{unique}'(x). \\
\hline
\end{array}
\end{array}
$$

Notice how the choices get reduced under merge. Now, if we merge `die Mutter` and `sieht`, `die Mutter` could in principle be either the subject or the object. However, in this merge it must inevitably be the object, since going from right to left in the argument structure, it matches the rightmost entry first, which corresponds to the object. So, it will end up being the object. After that, `die Katze` merges, but the object argument has been cancelled, so it becomes the subject instead.

(2.89)
$$
\begin{array}{c}
\texttt{/sieht/} \\
\begin{array}{|l|}
\hline
\langle e : \triangle\bigcirc : - \rangle, \\
\langle x : \nabla\oslash : \left[\begin{array}{l}\textsc{num}: sg \\ \textsc{case}: nom\end{array}\right] \\
\langle y : \nabla\oslash : [\textsc{case} : acc] \rangle. \\
\hline
e \\
\hline
\mathsf{see}'(e); \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq y. \\
\hline
\end{array}
\end{array}
$$

This means that although both word orders, (2.80) and (2.90), are allowed in German, (2.80) and (2.90) do not mean the same. In (2.90), the subject is `die Mutter` and `die Katze` is object.

(2.90) `..., dass die Katze die Mutter sieht.`
..., that the cat the mother sees.

(2.91) `..., dass die Mutter die Katze sieht.`
..., that the mother the cat sees.

This suggests that the second option is incorrect for German. However, we briefly note that the assumption that all word orders are equally grammatical is not justified, but within the present system the differences between the word orders cannot be brought to light. An example has been noted in [69]:

(2.92) `Maria mischt Wasser Wein bei.`
Maria mixes wine into water.

(2.93) `Maria mischt Wein Wasser bei.`
Maria mixes water into wine.

The NPs `Wasser` and `Wein` do not show the case distinction between dative and accusative. These two sentences indeed are not synonymous, as the glosses indicate. If we add the determiners, however, this effect disappears:

(2.94) `Maria mischt das Wasser dem Wein bei.`

(2.95) `Maria mischt dem Wein das Wasser bei.`
Maria mixes wine into water.

(2.96) `Maria mischt den Wein dem Wasser bei.`

(2.97) `Maria mischt dem Wasser den Wein bei.`
Maria mixes water into wine.

Let us summarize the possibilities that we have with argument structures. First, languages can use E-access or G-access. Suppose a language uses E-access. Then this language is fully structural. The argument structure of the head is projected uniquely into the syntax. The only parameters left are the word order parameters. If we set them uniformly right or left, we get SOV, OSV, VOS and VSO languages. Notice however that in VSO and OSV languages, the verb forms a constituent with its subject. (Although it goes against many currently accepted analyses, it has been claimed for languages like Berber and Toba-Batak by Keenan in [54] that these languages are VSO and that VS is a constituent.) For OVS and SVO we simply let the verb pick different directionality for subject and object. However, we may also leave the directionality unspecified either partially or with both arguments. This generates a few patterns that are to our knowledge not attested (eg if the subject is on either side but the object to the right, this language will allow for SVO/VOS patterns). However, just in case both subject and object are not directionally fixed, we get a language that specifies only immediate dominance but not linear precedence. This has been argued for by Staal for Sanskrit word order patterns (see [36]). This means that if verb forms a constituent with its object (as is generally assumed) we get the following alternative word orders:

(2.98) [S [V O]], [S [O V]], [[V O] S], [[O V] S]

If G-access is allowed, we get languages that differ from the previous languages only in that they allow for scrambling. If the verb is at the right periphery, that is, all arguments are to the left, then we get German type clause structure. With all arguments looked for to the right, we get the mirror image of German. If directionality is unspecified, we get Latin, as discussed above. It is worthwile pointing out that the present model, although allowing free (or freer) word order, nevertheless has a notion of canonical word order. This is so since the argument structure of the verb is a sequence, not a set. And these languages allow for alternate word orders independently of syncretism (under the first option). Case syncretism increases pressure to conform to default word order but it is difficult to state in exact terms how stringent this requirement is. For example fact, since in German the nominative and accusative are morphologically distinct only in the masculine singular, we find that there is a general bias against OSV constructions. Yet, number agreement also helps. In the following sentence the singular agreement `hat` in combination with the fact that `die größten Kritiker` is definitely plural makes it clear right away that the subject is yet to come, and that `die größten Kritiker` is actually the object.

(2.99) `Die größten Kritiker hat der Papst zu Hause.`
the biggest critics has the pope at home
The pope's biggest critics are in his home country.

Furthermore, in languages with rich morphology the word order freedom tends to be used to encode other features, in particular topic and focus. If we include prosodically marked discourse relations into the feature system of the DPs, then we can account for the fact that in German scrambled elements must be marked for discourse relations. Furthermore, we may restrict freedom of access in such a way that it is sensitive to certain features and not others. We shall not explore this further, however.

Notes on this section. Word order can be freed up even more if we allow fusion. Then we get languages that look more like Australian languages, such as Dyirbal and Jiwarli, in which there is not even a DP constituent. However, fusion is quite powerful and one must carefully look into the facts here. Notice that Dyirbal and Jiwarli do not show any sign of a canonical word order, in which case it would be justified to assume that the argument structures are sets rather than sequences. However, if that is so, then there can be no verbs assigning two accusatives to two arguments. This must be examined carefully.

We also note that if full fusion is allowed in syntax then even under the assumption of G-access we do not get fully free word order. Here is an example, which is reminiscent of the Golden Line in Latin verse. Suppose that a head H looks for two complements, C_1 and C_2, and that each complement is again looking for a complement. That is to say, C_1 has D_1 as complement and C_2 has D_2 as its complement. Then $HC_2C_1D_1D_2$ is accepted, while $C_1D_2HD_1C_2$ is not. The reason is that H cannot combine with either D_1 or D_2 even under fusion. But neither can C_1 combine with D_2 or C_2 with D_1. [9] This means that the theory remains restrictive with respect to word order. Whether or not these word orders turn out to be the right ones remains to be seen.

2.7 Under- and Overexposure

By design, the present theory also intends to cover agreement morphology. This means that it does not assume that the Hungarian word `tanáromnak` (teacher-POSS:1ST.SG-DAT=‘to my teacher’) is unanalysable. Rather, it assumes that this word is composed from three parts in the same way as it does for its English translation. We shall illustrate the way this is achieved in more detail in the next chapter. Here

[9] This argument must be carefully constructed. If H selects its arguments under the name α and β, while C_1 selects its complement under γ and C_2 selects D_2 under δ and if α, β, γ and δ are pairwise distinct, then no parse exists. Otherwise, if for example $\gamma = \delta$, then C_1D_2 and C_2D_1 are constituents. What we are saying in this case is that $HC_2C_1D_1D_2$ cannot be parsed to mean the same as $(H(C_1D_1))(C_2D_2)$.

we shall address a fundamental problem that such an approach faces. Consider the following German NP:

(2.100) `den schwarzen Katern`
the-DAT.PL black-DAT.PL tomcat-DAT.PL

The morphological paradigms of determiners, adjectives and nouns in German are quite distinct. In the singular, the noun shows no case distinction, but in the plural the dative takes `n`. Thus, we may assume that all three have been formed from a root that has no case and no number by adding successively a number and a case morpheme. (Determiners and adjectives also get a gender morpheme.) Let us concentrate on number. Plural has a meaning; whereas the singular denotes one of its kind, plural denotes a group (of more than one). If we were to assume that the plural morpheme contributes its meaning every time it is added to a word we face several problems. The first, harmless one, is that we keep iterating the same meaning. The second, less harmless one, is that everytime we add the plural we get a different meaning. Roughly, the addition of plural to `schwarz` ('black') will result in an adjective that denote a group of black things, while the addition to `Kater` ('tomcat') will result in a noun denoting a group of tomcats. In sum, the above will say that we are dealing with a group of things that is a group of black things in addition to being a group of cats. This will not work when the adjective is not intersective. For example, a group of big mice is not necessarily a group of big things and a group of mice. Thus the iteration of the plural meaning creates partly confusion and partly incorrect meanings. We should establish in detail where exactly the plural meaning comes into being.

In the life of an agreement feature we can establish three locations: the word which consumes the feature (by selecting that feature as part of its argument), the words which transmit the feature (for example adjuncts) and finally the word which carries the feature as part of its argument structure of a variable that is neither head nor adjunct. For example, the Latin sentence contains four occurrences of the plural morpheme each associated with the same object.

	`cantant`	`tres`	`magni`	`consules`
(2.101)	sing-3.PL	three-PL	big-PL	consul-PL
	$\triangledown$	$\lozenge$	$\lozenge$	$\vartriangle$

The noun brings the variable into existence and with it the feature 'plural', and the adjective and the numeral each transport the feature until it ends in the verb. All these four occurrences must be considered as part of one meaning expressed (we also say **exposed**) four times.

The really crucial points are the begin and the end of the chain. In fact, at both ends interesting things happen. First, it turns out that some elements select a feature without there being a clear meaning associated with it. The clearest example is that of case selection. The accusative by itself can mean a few things, for example a stretch of time. On the other hand, if a verb selects accusative this meaning is absent. In fact it is safe to say that there is no meaning associated with the accusative whatsoever. The feature has become purely formal. Similarly, there are words which are in the plural but the meaning is nevertheless singular; an example is `scissors` (which is singular in German). This means that either point can block the meaning of the morpheme to come into play. The noun can resist it and the head can as well. The problem that this poses is that we must account for the fact that the plural meaning is completely absent if it is blocked by either the noun or the verb that consumes it as in.

(2.102) `The scissors are on the table.`

We shall come back to this problem later. Principally, the solution lies in the following. We assume that the fate of an agreement morpheme is decided at the end when it is consumed as part of an argument. Until then it is considered purely formal, as an agreement device. This means in practice that even if the morpheme has meaning, the following mergers will not result in any addition of meaning: adding the feature to an adjunct variable or adding the feature to an argument variable.

This can be achieved by splitting the agreement morpheme into at least two morphemes, only one of which carries meaning. There will be a plural agreement suffix for adjectives, one for nouns, and one for verbs. It will turn out that in part this split is justified. In languages where the verb overtly agrees with two of its arguments we need to distinguish subject plural agreement from object plural agreement. However, the distinction between plural agreement on the adjective and plural agreement on the noun is moot; for they are often even formally similar and this unity needs to be explained.

2.8 Infinitives and Complex Predicates

Infinitives provide interesting evidence that fusion is a real option of grammar. Recall that three very closely related languages, English, Dutch and German, behave very differently with respect to embedded infinitives. We give an example. (2.103) can be translated by (2.104) into German and by (2.105) into Dutch.[10]

(2.103) `I said that Karl saw Peter let Mary teach the children to swim.`

(2.104) `Ich sagte, dass Karl Peter Maria die Kinder schwimmen lehren lassen sah.`

(2.105) `Ik zei dat Karl Peter Maria de kinderen zag laten leren zwemmen.`

The patterns are as follows.

	English:	$\ldots NP_1\ V_1\ NP_2\ V_2 NP_3\ V_3\ NP_4\ V_4$
(2.106)	German:	$\ldots NP_1\ NP_2\ NP_3\ NP_4\ V_4\ V_3\ V_2\ V_1$
	Dutch:	$\ldots NP_1\ NP_2\ NP_3\ NP_4\ V_1\ V_2\ V_3\ V_4$

Here, NP_i is the subject of the infinitive V_i (and the object of V_{i-1} for $i > 1$). These dependencies are trans-context free. However, for semantic purposes the weak generative capacity is not enough; we need the correct structures as well. It is not hard to show that the fusion free calculus is context free. This provides an abstract argument why fusion is needed. We shall show in this section that it can actually also provide a correct analysis of all three languages.

Before we begin the discussion, we shall remark that infinitives differ from finite verbs in that they do not assign case to their subject argument. This means that the subject of the infinitive must be expressed in a higher clause, since in these languages overt NPs require case. So, while `John swims` is a well-formed sentence, since `John` actually has case and so can be the argument of swims, in `John to swim` this is not the case and the sentence is ungrammatical. In the sentence `Mary asked John to swim` the constituent `John` actually has accusative case,

[10]There are—at least in German—many different ways to express (2.103), of which (2.104) is one.

which we can demonstrate by exchanging it for a pronoun. We have

(2.107) `He swims.`

(2.108) *`Mary asked he to swim.`

(2.109) `Mary asked him to swim.`

Therefore, `John` occupies the object position of the verb `asked` and not the subject position of the verb `to swim`. In the analysis below infinitives therefore do not assign any (nominative) case.

We start with the English construction. There are two types of verbs, basic verbs such as `swim` and serial verbs such as `let`. The latter take an NP and an infinitive as a complement and require that the NP is the subject of the embedded infinitive. Therefore the semantics of `swim`, `Mary` and `let` is as follows. (We have simplified the representation of the AVSs.)

(2.110)

/swim/
$\langle e : \triangle \bigcirc : \text{INF} \rangle$
e
swim′(e); act′$(e) \doteq x$.

/Mary/
$\langle x : \triangle \bigcirc : \text{ACC} \rangle$
x
$x \doteq m$.

/let/	
$\langle e : \triangle \bigcirc : \text{INF} \rangle$, $\langle f : \triangledown \oslash : \text{INF} \rangle$,	$\langle y : \triangledown \oslash : \text{ACC} \rangle$
e, f	
let′(e);	act′$(e) \doteq x$;
thm′$(e) \doteq f$;	ben′$(e) \doteq y$;
act′$(f) \doteq y$.	

So, there is an event e of letting, and its theme (that which is let to be the case) is f. The beneficiary of the letting is y; y is also the actor of f.[11] So, y is doing double duty: it is the beneficiary of the letting event but the actor of the embedded event. This is desirable since it allows to incorporate the distinction between subject control and object control of infinitives. We exemplify this with the verbs

[11] It may be questionable to assume that verbs selecting an event actually also involve existential quantification over that event. The word `let` might only mean that there is an event of letting but not necessarily that there is an event that is being let to be the case. For example, if I let someone enter my room, he can still decide not to enter at all. There is however still the event of me letting him enter my room. So, we might decide not to put f into the upper box. Similarly with `persuade`, where the secondary event might actually be in the future as in `persuade to go to London`. Intensional verbs are still different. Eventually, this must be resolved by appeal to parameters, see Chapter 4. Nothing hinges on the analysis given above, however.

`promise` and `persuade`.

(2.111)

$$\begin{array}{|ll|}
\multicolumn{2}{c}{\texttt{/promise/}} \\
\hline
\langle e : \vartriangle \bigcirc : \textsc{inf} \rangle, & \\
\langle f : \triangledown \ominus : \textsc{inf} \rangle, & \langle y : \triangledown \ominus : \textsc{acc} \rangle. \\
\hline
e, \quad f & \\
\hline
\mathsf{promise}'(e); & \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq y; \\
\mathsf{act}'(f) \doteq x. & \\
\hline
\end{array}$$

(2.112)

$$\begin{array}{|ll|}
\multicolumn{2}{c}{\texttt{/persuade/}} \\
\hline
\langle e : \vartriangle \bigcirc : \textsc{inf} \rangle, & \\
\langle f : \triangledown \ominus : \textsc{inf} \rangle, & \langle y : \triangledown \ominus : \textsc{acc} \rangle. \\
\hline
e, \quad f & \\
\hline
\mathsf{persuade}'(e); & \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq f; & \mathsf{pat}'(e) \doteq y; \\
\mathsf{act}'(f) \doteq y. & \\
\hline
\end{array}$$

`Promise` has two arguments besides the subject (x), namely the beneficiary, y and the infinitive. `Persuade` differs only in the thematic role of y; here it is a patient, but this is insignificant for the present purposes. Now, while the actor of the complement f is x in the case of `promise`, it is y in the case of `persuade`. Therefore, with this semantics, for (2.113) it turns out that it is Albert who will do the theorem proving and that in (2.114) it is Jan. This is as it should be.

(2.113) `Albert promises Jan to prove new theorems.`

(2.114) `Albert persuades Jan to prove new theorems.`

A problematic aspect of the present analysis is the fact that it assumes that the subject of the lower infinitive is the actor; it must do so in order to identify the subject. There is an alternative solution which makes use of polyadic merge.

(2.115)

`/swim/`

$$\begin{array}{|l|}\hline \langle e : \triangle\bigcirc : \textsc{inf}\rangle \\ \langle x : \triangle\bigcirc : [\textsc{case} : \star]\rangle \\ \hline e \\ \hline \mathsf{swim}'(e); \\ \mathsf{act}'(e) \doteq x. \\ \hline \end{array}$$

`/let/`

$$\begin{array}{|ll|}\hline \langle e : \triangle\bigcirc : [\textsc{agr} : \mathit{inf}]\rangle, & \\ \langle x : \triangle\bigcirc : [\textsc{case} : \star], & \\ \langle y : \triangledown\ominus : [\textsc{case} : \star]\rangle, & \\ \langle f : \triangledown\ominus : [\textsc{agr} : \mathit{inf}]\rangle, & \\ \langle z : \triangledown\ominus : [\textsc{case} : \textsc{acc}]\rangle. & \\ \hline e, \quad f & \\ \hline \mathsf{let}'(e); & \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq z; \\ y \doteq z. & \\ \hline \end{array}$$

Here, `let` can be merged with `Mary` if the latter carries accusative case, and the result is

(2.116)

`/let Mary/`

$$\begin{array}{|ll|}\hline \langle e : \triangle\bigcirc : [\textsc{agr} : \mathit{inf}\rangle, & \\ \langle x : \triangle\bigcirc : [\textsc{case} : \star], & \\ \langle y : \triangledown\ominus : [\textsc{case} : \star]\rangle, & \\ \langle f : \triangledown\ominus : [\textsc{agr} : \mathit{inf}]\rangle, & \\ \hline e, \quad f & \\ \hline \mathsf{let}'(e); & \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq z; \\ z \doteq m; & y \doteq z. \\ \hline \end{array}$$

Next we perform polyadic merge with `swim` and we get

(2.117)

`/let Mary swim/`

$$\begin{array}{|ll|}\hline \langle e : \triangle\bigcirc : [\textsc{agr} : \mathit{inf}]\rangle, & \\ \langle x : \triangle\bigcirc : [\textsc{case} : \star]\rangle, & \\ \hline e, \quad f & \\ \hline \mathsf{let}'(e); & \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq z; \\ z \doteq m; & \mathsf{swim}'(f); \\ \mathsf{act}'(f) \doteq y. & y \doteq z. \\ \hline \end{array}$$

The semantics comes out in the correct way. One may consider the fact that the same argument is imported twice (though the referents are technically different) unsound, but this is perfectly legal option.

Now notice the following. The verb `to let` demands three arguments, an infinitival complement, an object noun phrase and a subject noun phrase. If we allow the arguments to be accessed freely then the object NP and the infinitive can be freely permuted. This means that the sentence (2.118) would be grammatical, contrary to fact.

(2.118) *`I said that John lets swim Mary.`

Fortunately, the conditions of E-access rule this out. If there is an argument that is prior to the sentential complement, then that argument must be merged away first. The only grammatical constructions are therefore those in which the serial verbs take first a complement NP to the right, and then an infinitival complement. These are exactly the facts of English.

Now we turn to German. The major difference between German and English is the directionality of the selection. The verbs in German select both the noun phrase and the infinitive to their left. An additional difference is that the infinitive is selected first. The representations for `schwimmen` and `lassen` are therefore as follows.

(2.119)

$$\begin{array}{c} \texttt{/schwimmen/} \\ \begin{array}{|l|} \hline \langle e : \triangle\bigcirc : [\textsc{agr} : \mathit{inf}]\rangle. \\ \langle x : \triangle\bigcirc : [\textsc{case} : \star]\rangle \\ \hline \hline e \\ \hline \mathsf{swim}'(e); \\ \mathsf{act}'(e) \doteq x. \\ \hline \end{array} \end{array}$$

(2.120)

$$\begin{array}{c} \texttt{/lassen/} \\ \begin{array}{|ll|} \hline \langle e : \triangle\bigcirc : [\textsc{agr} : \textsc{inf}]\rangle, & \langle z : \triangle\bigcirc : [\textsc{case} : \star]\rangle, \\ \langle y : \nabla\obslash : [\textsc{case} : \textsc{acc}], \rangle, & \langle x : \nabla\obslash : [\textsc{case} : \star], \\ \langle f : \nabla\obslash : [\textsc{agr} : \textsc{inf}]\rangle. & \\ \hline \hline e, \quad f & \\ \hline \mathsf{let}'(e); & \mathsf{act}'(e) \doteq z; \\ \mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq y; \\ x \doteq y. & \\ \hline \end{array} \end{array}$$

Notice that German `lassen` takes one more argument than English `let`. This is necessitated by the different syntax. The German infinitive first merges polyadically with the infinitive; at this stage the variable x is identified. Only after that the accusative object is taken in.

For German, too, we must posit a restriction on access, this time however for the object referents. They are not allowed to skip an event referent. Otherwise (2.104) would be grammatical just like (2.122).

(2.121) *`Ich sagte, dass Karl schwimmen Peter ließ.`
I said that Karl swim Peter let.

(2.122) `Ich sagte, dass Karl Peter schwimmen ließ.`
I said that Karl Peter swim lets.

The syntactic data of German are complicated by many factors, one being that auxiliaries and verbs do not let an infinitive appear on the right hand side, but many other verbs do. One such verb is `helfen`. Another complication is that while the infinitive appears on the left hand side, the finite clause complements are typically on the right side. However, these additional facts can be incorporated by suitable changes in the argument structure. Before we can address that we shall discuss the case of Dutch.

In Dutch, the facts get rather involved. First of all, from abstract arguments we know that merge alone would not allow to generate the Dutch data. This is so because Dutch is not strongly context free. If we allow only the merge, then we have only finitely many rules, each of which are context free. Moreover, the syntactic relations are mirrored by the semantic relations in a rather straightforward way. The subject and object of a verb must be within the extended projection of the verb. Hence, we must allow for fusion of argument structures. Specifically, we allow two verbs to fuse their argument structures. This generates a structure that is quite similar to the ones found in the literature (GB and LFG). The verbs join into a big cluster and only after that the NP arguments are discharged, one after the other. To account for these facts we do the following. We introduce two kinds of vertical diacritics, one for merge and another for fusion. So, if an argument is selected for, the functor can choose whether or not it selects through fusion. We write $\blacktriangledown$ and $\blacklozenge$ if fusion is required and $\triangledown$ and $\lozenge$ otherwise.[12]

[12]We could also introduce a third kind for those heads that allow both merge and fusion for their arguments (as is the case with German verbs) but this can also be accounted for by using different lexical entries. Thus we refrain from intoducing more symbolism here.

The semantics of the verbs `zwemmen` and `laten` are now as follows.

(2.123)

$$
\begin{array}{|l|}
\multicolumn{1}{c}{\texttt{/zwemmen/}} \\
\hline
\langle e : \triangle\bigcirc : [\textsc{agr} : \mathit{inf}]\rangle. \\
\langle x : \triangle\bigcirc : [\textsc{case} : \bigstar]\rangle. \\
\hline
e \\
\hline
\mathsf{swim}'(e); \\
\mathsf{act}'(e) \doteq x. \\
\hline
\end{array}
$$

(2.124)

$$
\begin{array}{|ll|}
\multicolumn{2}{c}{\texttt{/laten/}} \\
\hline
\multicolumn{2}{|l|}{\langle e : \triangle\bigcirc : [\textsc{agr} : \mathit{inf}]\rangle,} \\
\multicolumn{2}{|l|}{\langle x : \triangle\bigcirc : [\textsc{case} : \bigstar]\rangle,} \\
\multicolumn{2}{|l|}{\langle y : \triangledown\olessthan : [\textsc{case} : \mathit{acc}]\rangle,} \\
\multicolumn{2}{|l|}{\langle z : \triangledown\bigcirc : [\textsc{case} : \bigstar]\rangle,} \\
\multicolumn{2}{|l|}{\langle f : \blacktriangledown\ogreaterthan : [\textsc{agr} : \mathit{inf}]\rangle.} \\
\hline
e, \quad f & \\
\hline
\mathsf{let}'(e); & \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq y; \\
\mathsf{act}'(f) \doteq y; & y \doteq z. \\
\hline
\end{array}
$$

Now the verb is looking first for an event referent to the right, and then for an object referent to its left. The rule for fusion is as follows. The entire argument structure minus the first entry of the argument is inserted in place of the referent that it identifies with; the identified referent is cancelled. In polyadic fusion all mergers are performed before the remainder is added to the list. (The precise details of the serialisation do not matter throughout this book.) The serial verb `zag laten` is generated through fusion: [13]

(2.125)

$$
\begin{array}{|ll|}
\multicolumn{2}{c}{\texttt{/zag laten/}} \\
\hline
\langle e : \triangle\bigcirc : [\textsc{agr} : \texttt{past}\rangle, & \langle x : \triangle\bigcirc : [\textsc{case} : \bigstar]\rangle, \\
\langle y : \triangledown\olessthan : [\textsc{case} : \mathit{acc}]\rangle, & \langle z : \triangledown\olessthan : [\textsc{agr} : \mathit{acc}]\rangle, \\
\multicolumn{2}{|l|}{\langle f : \blacktriangledown\ogreaterthan : [\textsc{agr} : \mathit{inf}]\rangle.} \\
\hline
e, \quad f & \\
\hline
\mathsf{see}'(e); & \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq f; & \mathsf{ben}'(e) \doteq y; \\
\mathsf{act}'(f) \doteq y; & \mathsf{let}'(f); \\
\mathsf{thm}'(f) \doteq f; & \mathsf{act}'(f) \doteq z; \\
y \doteq z. & \\
\hline
\end{array}
$$

[13] To illustrate how the variables are tracked through fusion we use here the annotation by superscripts.

Now notice that the lexemes for Dutch are different from the German ones in that they select the infinitive to the right, and they are also different from the English ones because the nominal arguments are consistently to the left. If Dutch has the extra option of fusion then not only would those sentences be grammatical which use fusion but also those which can be obtained through standard merge. The following sentences would be grammatical (with the meaning being that of (2.105)).

(2.126) *`Ik zei dat Karl Peter Maria [zag laten] de kinderen [leren zwemmen].`

(2.127) *`Ik zei dat Karl Peter zag [Maria de kinderen [laten leren zwemmen]].`

(2.128) *`Ik zei dat Karl Peter zag [Maria laten [de kinderen [leren zwemmen]]].`

Namely, `leren zwemmen` is a one-place predicate taking an argument to the left (the one who is being taught). This argument is consumed to the left giving rise to a zero-place predicate `de kinderen leren zwemmen`. This shows why (2.128) is generated by the calculus. Likewise, `laten leren zwemmen` will be a two-place predicate taking as first argument to the left the one who is being taught and secondly the one who is being let to do the teaching. This explains the sentence (2.127). (2.126) is generated as follows. `Zag laten` is a three-place predicate taking an infinitival complement, in this case `de kinderen leren zwemmen`. We shall stress that fusion is not a global option for Dutch. So, it is not generally the case that Dutch allows fusion in contrast to English. Rather, it is specific arguments that allow for fusion in contrast to others.

It has been argued that German verbs too trigger fusion (even though you cannot see that by looking at our examples). So, the lexical entry for `lassen` is now as follows.

(2.129)

`/lassen/`

$\langle e : \vartriangle \bigcirc : [\textsc{case} : inf] \rangle$,	$\langle x : \vartriangle \bigcirc : [\textsc{case} : \star] \rangle$,
$\langle y : \triangledown \olessthan : [\textsc{case} : acc] \rangle$,	$\langle z : \triangledown \olessthan : [\textsc{case} : \star] \rangle$,
$\langle f : \blacktriangledown \olessthan : [\textsc{agr} : \textsc{inf}] \rangle$.	
$e, \quad f$	
$\mathsf{let}'(e)$;	$\mathsf{act}'(e) \doteq x$;
$\mathsf{thm}'(e) \doteq f$;	$\mathsf{ben}'(e) \doteq y$;
$\mathsf{act}'(f) \doteq y$.	$y \doteq z$.

We shall agree the following: if an argument is selected through fusion, it must be selected first. (Moreover, there shall be at most one argument that can be selected through fusion.) Why this is so will be discussed in Section 3.7. This generalizes the restriction we have made with respect to access in the German verb.

So, on what grounds are (2.126) – (2.126) excluded? We shall say in addition that fusion is restricted (in Dutch) to words and moreover it produces only words. So, `laten` selects only words through fusion, and when `laten` and `zwemmen` merge, the result is again a word. We shall show in Section 3.7 how this can be implemented into the argument structure. Then the examples (2.126) – (2.128) are excluded. To see this, look at the argument structure of `zag laten` in (2.125). The f^2 argument is inherited from `laten`. Since `laten` is in turn a raising verb, this argument is identified through fusion. This means that it must be a word and it must be the first that is identified. Hence, it can neither combine with `de kinderen`, since this is not an event, nor with `de kinderen leren zwemmen`, since that is not a word. The same arguments work for (2.127) and (2.128).

Therefore, this analysis gets at least the basic syntactic structure right. Let us now turn to word order variation in Dutch and German. In Dutch there is next to no morphological variation, and so the arguments may not be permuted. Therefore, (2.130) and (2.131) cannot be taken to mean the same as (2.105).

(2.130) `Ik zei dat Peter Karl Maria de kinderen zag laten leren zwemmen.`

(2.131) `Ik zei dat Karl de kinderen Peter Maria zag laten leren zwemmen.`

The same holds for German. However, in those cases where there is a morphological differentiation, alternative word orders are allowed. So, (2.132) – (2.134) all mean the same as (2.135).

(2.132) `Ich sagte Karl, dass ich ihr den Kühlschrank zu reparieren versprochen hatte.`

(2.133) `..., dass den Kühlschrank ich ihr zu reparieren versprochen hatte.`

(2.134) `..., dass ihr den Kühlschrank ich zu reparieren versprochen hatte.`

I told Karl, that I promised her to repair the refrigerator.

These examples show that we are really dealing with a complex predicate here (or, following traditional usage, we have a phenomenon of clause union). For the arguments can be serialized differently exactly when they exhibit clear morphological differentiation with respect to the names in question.

We close by noting that German allows even freer word order than permitted by the present system. Notably, infinitives are allowed to consume their case marked arguments before they fuse into a complex predicate.

(2.135) ..., dass ich [den Kühlschrank zu reparieren] [ihr versprochen hatte].

(2.136) ..., dass [den Kühlschrank zu reparieren] ich ihr versprochen hatte.

However notice that we are dealing here with another infinitive, namely the zu-infinitive, which might be responsible for this additional freedom. We shall not discuss this further. Notice that this is a feature of German. In Dutch, this phenomenon is absent. Lack of case marking would result in too much ambiguity. For then the highest raising verb can alternatively take the last NP as its object, rather than the first. Namely, in that case (2.105) can alternatively be rendered as (2.138).

(2.137) Ik zei dat Karl Peter Maria de kinderen [zag [laten leren zwemmen]].

(2.138) Ik zei dat Peter Maria de kinderen [[Karl zag] laten leren zwemmen].

Notes on this section. The idea that the complex verbs of German and Dutch form a cluster which functions as a single word, shows up in many other syntactic theories. In GB, the verbs are raised and adjoin to the raising head. This adjunction is a zero-level (=head-to-head) adjunction. Since zero-level means 'is a word for syntactic purposes', we get the distinction between the languages by parametrizing for the availability of raising and for the directionality of adjunction. Hence, fusion is like zero-level adjunction. We shall deal later with complex verbs in Chapter 6.5. The word order freedom in German raising constructions has been studied in [10]. It is claimed there that the construction exceeds the power of Linear Context Free Rewriting Systems (LCFRSs). The argument is based on the fact that clause union is not bounded. However, if we are right, then the order of arguments is nevertheless restricted by their overt morphology. Since there are only a finite number of cases to deal with, not all serializations of the arguments can go together with the same meaning. This does not affect the string language, though. In Section 6.5 it shall be argued that clause union is restricted by other factors, which will yield that it is not so free after all (and within the reach of LCFRSs).

2.9 Logical Connectives, Groups and Quantifiers

Merge as we have defined it so far is monotone: the reader may check that if we have structures $\mathfrak{S}_1$ and $\mathfrak{S}_2$, then the semantics underlying $\mathfrak{S}_1 \oplus \mathfrak{S}_2$ logially implies that underlying $\mathfrak{S}_1$ and $\mathfrak{S}_2$. Thus the semantics is incomplete: there is no way to implement a semantics of negation.[14] Meanings are therefore only added, there is no way to negate meanings, or quantify over objects. This is obviously not enough to cover the full range of natural language expressions. In this section we shall propose a new type of variable that will allow to deal with logical connectives and quantifiers. The problem with logical connectives (`and`, `or`, `not` and so on) is twofold: from a semantic point of view they do not take variables but the entire proposition in their scope; from a combinatorial point of view their syntax is very flexible. In general, they may take arguments of any type. Any constituent may be negated, any two constituents may be coordinated. The only restriction is that in the binary cases we may only take two constituents of the same type. It is not the place here to defend the correctness of this analysis. [53] have argued convincingly that any syntactic category forms a boolean structure. Moreover, the exceptions to the identity restriction tend to be marginal so that we simply disregard them.

Let us return to Section 2.2. We have outlined there how in DRT complicated logical structures are built up using various connectives such as :, $\cup$, $\vee$ and $\neg$. We will now consider how these connectives can be built into the present system.[15] The addition we are going to make is the following. First, in addition to standard variables ranging over objects of various types, there are also variables ranging over DRSs (or propositions). We write them $\boxed{1}$, $\boxed{2}$ and so on. Finally, there are

[14]This is not quite correct. We could do the following: ever formula φ is translated into $p \doteq \varphi$, and the semantics of negation is, for example, $q \doteq \neg p$. Using the calculus of parameters of Chapter 4 this can be implemented wasily. Unfortunately, this will not eliminate the problems of polymorphism.

[15]We stress here once again that the use of DRT has only pedagogical reasons. The technique can easily be recast in dynamic predicate logic if needed.

also variables over argument structures, denoted by gothic letters: $\mathfrak{x}$, $\mathfrak{y}$ and so on. In place of AVS α one may also write $\alpha\dagger\mathfrak{x}$ and in place of the single variable x one may write $x\dagger\boxed{1}$. Any of the two variables is optional. An argument identification statement may then assume any of the following nine forms

$$
(2.139)\quad
\begin{array}{l}
\langle x \; {:}\partial{:}\alpha\rangle \\
\langle x \; {:}\partial{:}\alpha\dagger\mathfrak{x}\rangle \\
\langle x \; {:}\partial{:}\mathfrak{x}\rangle \\
\langle\boxed{1} \; {:}\partial{:}\alpha\rangle \\
\langle\boxed{1} \; {:}\partial{:}\alpha\dagger\mathfrak{x}\rangle \\
\langle\boxed{1} \; {:}\partial{:}\mathfrak{x}\rangle \\
\langle x\dagger\boxed{1}{:}\partial{:}\alpha\rangle \\
\langle x\dagger\boxed{1}{:}\partial{:}\alpha\dagger\mathfrak{x}\rangle \\
\langle x\dagger\boxed{1}{:}\partial{:}\mathfrak{x}\rangle
\end{array}
$$

However, notice that this is just a notational simplification since a variable occurring in the argument section does not have to be part of the semantics. The fact that the variables are denoted differently than ordinary variables will prevent confusion.

The rules for merge change in the following way. If $\mathfrak{x}$ is present, μ can only merge as the functor, it is not an argument. The merge will succeed if the merge would succeed with $\mu' := \langle x : \partial : \alpha\rangle$ in place of μ. In this case, $\mathfrak{x}$ is bound to the entire argument structure of the argument, and $\boxed{1}$ is bound to the semantics underlying it.

Let us give examples. Here are argument structures for `and`, `or` and `not`.

$$
(2.140)\quad
\begin{array}{ll}
\texttt{/and/} & \texttt{/or/} \\
\begin{array}{|l|}\hline
\mathfrak{x}, \\
\langle x\dagger\boxed{1} : \nabla\olessthan : \mathfrak{x}\rangle, \\
\langle y\dagger\boxed{2} : \nabla\ogreaterthan : \mathfrak{x}\rangle. \\ \hline
\boxed{2}\cup\boxed{1} \\ \hline
\end{array}
&
\begin{array}{|l|}\hline
\mathfrak{x}, \\
\langle x\dagger\boxed{1} : \nabla\olessthan : \mathfrak{x}\rangle, \\
\langle y\dagger\boxed{2} : \nabla\ogreaterthan : \mathfrak{x}\rangle. \\ \hline
\boxed{2}\vee\boxed{1} \\ \hline
\end{array}
\\[2ex]
\texttt{/not/} & \\
\begin{array}{|l|}\hline
\mathfrak{x}, \\
\langle x\dagger\boxed{1} : \nabla\ogreaterthan : \mathfrak{x}\rangle. \\ \hline
\neg\boxed{1} \\ \hline
\end{array}
&
\end{array}
$$

(We remark here that $\boxed{2} \vee \boxed{1}$ must be a DRS with a head section. So, it is not simply the disjunction of two DRSs each with their own head section. It also creates a new main head section.) Here is the representation for `every`:

$$
(2.141)\quad
\begin{array}{|l|}
\multicolumn{1}{c}{\texttt{/every/}} \\
\hline
\langle y : \lozenge\ogreaterthan : \begin{bmatrix} \textsc{num}{:}sg \\ \textsc{cat}\ {:}ob \end{bmatrix} \rangle, \\
\langle x\dagger\boxed{1} : \lozenge\ogreaterthan : \begin{bmatrix} \textsc{num}{:}sg \\ \textsc{cat}\ {:}ob \end{bmatrix} \rangle. \\
\hline
\varnothing \\
\hline
y \doteq \{\{x\} : \boxed{1}(x)\} \\
\hline
\end{array}
$$

The variable x in the argument structure is the same as the x in the DRS, but it is bound there.

To see an easy example, we produce the Latin `non dat` (‘he does not give’). The argument structure of `dat` is:

$$
(2.142)\quad
\begin{array}{|ll|}
\multicolumn{2}{c}{\texttt{/dat/}} \\
\hline
\multicolumn{2}{|l|}{\langle e : \vartriangle\bigcirc : \begin{bmatrix} \textsc{pers}{:}3 \\ \textsc{num}{:}sg \end{bmatrix} \rangle,} \\
\multicolumn{2}{|l|}{\langle x : \triangledown\olessthan : \begin{bmatrix} \textsc{case}{:}nom \\ \textsc{num}{:}sg \end{bmatrix} \rangle,} \\
\multicolumn{2}{|l|}{\langle y : \triangledown\ogreaterthan : [\textsc{case} : acc] \rangle,} \\
\multicolumn{2}{|l|}{\langle z : \triangledown\ogreaterthan : [\textsc{case} : dat] \rangle.} \\
\hline
\multicolumn{2}{|l|}{e} \\
\hline
\mathsf{give}'(e); & \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq y; & \mathsf{ben}'(e) \doteq z; \\
\mathsf{time}(e) \doteq \mathsf{now}'. & \\
\hline
\end{array}
$$

Now, the variable $\mathfrak{x}$ can match any argument structure, in particular the one for `dat` and we get:

(2.143)

`/non dat/`

$$\left\langle e : \triangle\bigcirc : \begin{bmatrix} \text{PERS}:3 \\ \text{NUM}:sg \end{bmatrix} \right\rangle,$$
$$\left\langle x : \triangledown⧀ : \begin{bmatrix} \text{CASE}:nom \\ \text{NUM}:sg \end{bmatrix} \right\rangle,$$
$$\langle y : \triangledown⧁ : [\text{CASE} : acc] \rangle,$$
$$\langle z : \triangledown⧁ : [\text{CASE} : dat] \rangle.$$

$\neg$

e	
give′(e);	act′(e) $\doteq$ x;
thm′(e) $\doteq$ y;	ben′(e) $\doteq$ z;
time′(e) $\doteq$ now′.	

(We omit boxes around single entries.) So, we get the same argument structure again. The reason is that when non merges with `dat`, the variable $\mathfrak{x}$ is instantiated to the argument structure of `dat`. Since non also exports $\mathfrak{x}$, and $\mathfrak{x}$ is now instantiated to the argument structure of `dat`, this is the resulting argument structure.

When we approach the other connectives in the same way, we meet a small problem. `And`, for example, will take a complement C to its right, and $\mathfrak{x}$ will be instantiated to the argument structure of C. Subsequently, `and`$^\frown C$ looks to its left for an element with identical argument structure. However, the variables in the argument structure are part of the name of $\mathfrak{x}$, so if by chance D has the same argument structure as C with the variables being named differently, the merge will not succeed. Here is a simple example:

(2.144)

$$
\begin{array}{|l|}
\multicolumn{1}{c}{\texttt{/venit/}} \\
\hline
\langle e : \vartriangle \bigcirc : \begin{bmatrix} \textsc{cat} & : ev \\ \textsc{pers} & : 3 \\ \textsc{num} & : sg \end{bmatrix} \rangle, \\
\langle x : \triangledown \text{⦶} : [\textsc{case} : nom] \rangle. \\
\hline
e \\
\hline
\mathsf{come}'(e); \mathsf{act}'(e) \doteq x. \\
\hline
\end{array}
\quad \bullet \quad
\begin{array}{|l|}
\multicolumn{1}{c}{\texttt{/et/}} \\
\hline
\mathfrak{x}, \\
\langle \boxed{1} : \triangledown \text{⧀} : \mathfrak{x} \rangle, \\
\langle \boxed{2} : \triangledown \text{⧁} : \mathfrak{x} \rangle. \\
\hline
\boxed{2} \cup \boxed{1} \\
\hline
\end{array}
$$

$$
\bullet \quad
\begin{array}{|l|}
\multicolumn{1}{c}{\texttt{/vidit/}} \\
\hline
\langle e : \vartriangle \bigcirc : \begin{bmatrix} \textsc{cat} & : ev \\ \textsc{pers} & : 3 \\ \textsc{num} & : sg \end{bmatrix} \rangle, \\
\langle y : \triangledown \text{⦶} : [\textsc{case} : nom] \rangle. \\
\hline
e \\
\hline
\mathsf{see}'(e); \mathsf{act}'(e) \doteq x. \\
\hline
\end{array}
$$

For example, we may make the following choice for $\mathfrak{x}$:

(2.145)
$$
\mathfrak{x} := \begin{bmatrix} \langle e : \vartriangle \bigcirc : \begin{bmatrix} \textsc{cat} & : ev \\ \textsc{pers} & : 3 \\ \textsc{num} & : sg \end{bmatrix} \rangle, \\ \langle y : \triangledown \text{⦶} : [\textsc{case} : nom] \rangle. \end{bmatrix}
$$

Only with this choice, the last two can structures can merge and we get the following result:

$$
\begin{array}{c}
\texttt{/venit/} \\
\boxed{\begin{array}{l}
\langle e : \triangle\bigcirc : \begin{bmatrix} \textsc{cat} : ev \\ \textsc{pers} : 3 \\ \textsc{num} : sg \end{bmatrix} \rangle, \\
\langle x : \nabla\obar : [\textsc{case} : nom]\rangle.
\end{array}} \\
\boxed{\begin{array}{l} e \\ \hline \mathsf{come}'(e); \mathsf{act}'(e) \doteq x. \end{array}}
\end{array}
$$

(2.146)

$$
\bullet\quad
\begin{array}{c}
\texttt{/et vidit/} \\
\boxed{\langle \boxed{1}, \triangle\ \ominus : \begin{bmatrix} \langle e : \triangle\bigcirc : \begin{bmatrix} \textsc{cat} : ev \\ \textsc{pers} : 3 \\ \textsc{num} : sg \end{bmatrix} \rangle, \\ \langle y : \nabla\obar : [\textsc{case} : nom]\rangle \end{bmatrix} \rangle} \\
\boxed{\boxed{1} \cup \boxed{\begin{array}{l} e \\ \hline \mathsf{see}'(e); \mathsf{act}'(e) \doteq y \end{array}}}
\end{array}
$$

These two structures cannot merge, since $\boxed{1}$ is identified under a different argument structure, namely the following

(2.147)
$$
\mathfrak{x} := \begin{bmatrix} \langle e : \triangle\bigcirc : \begin{bmatrix} \textsc{cat} : ev \\ \textsc{pers} : 3 \\ \textsc{num} : sg \end{bmatrix} \rangle, \\ \langle x : \nabla\obar : \textsc{case} : nom]\rangle. \end{bmatrix}
$$

But this choice is in conflict with the requirement of the third argument structure. The problem is the choice of the variable names, which now have become part of the name of the referent $\boxed{1}$. For our present purposes the following can be done. Say that α and β **match**, in symbols $\beta \approx \alpha$, if β results from α by replacing uniformly certain variables. (The naming system of Chapter 5.6 does away with this problem to a large extent, though not completely.) We now define the merge with respect to second order argument structures as follows. $\langle\boxed{1} : \nabla\flat : \alpha\rangle$ identifies β (in the direction $\flat$) if $\beta \approx \alpha$. The merge is as follows.

(2.148)
$$
\begin{array}{c}\boxed{\begin{array}{l} \omega, \\ \langle \delta : \nabla\ominus : \alpha\rangle \end{array}} \\ \boxed{\phi(\delta)} \end{array}
\quad\bullet\quad
\begin{array}{c}\boxed{\beta} \\ \boxed{\theta}\end{array}
\quad=\quad
\begin{array}{c}\boxed{\omega} \\ \boxed{\phi(\theta^{\sigma})}\end{array}
$$

Here, σ is a substitution such that $\beta^\sigma = \alpha$. Hence, under these renewed definitions the above merge can be carried out and we get

(2.149)
$$\begin{array}{|l|}\multicolumn{1}{c}{\texttt{/venit et vidit/}}\\ \hline \langle e : \triangle\bigcirc : \begin{bmatrix} \textsc{cat} : ev \\ \textsc{pers} : 3 \\ \textsc{num} : sg \end{bmatrix} \rangle, \\ \langle y : \triangledown\oplus : [\textsc{case} : nom] \rangle. \\ \hline \varnothing \\ \hline \mathsf{come}'(e); \mathsf{act}'(e) \doteq x; \mathsf{see}'(e). \\ \hline \end{array}$$

A note is in order on the possible values of $\mathfrak{z}$. Since we do not use names but descriptions of names, we can have descriptions that are partial, for example, by not restricting the directionality of the complement. Hence, we shall finally say that $\langle \delta : \triangledown\flat : \alpha \rangle$ **identifies** β in the direction $\flat$ if there is a substitution σ such that $\alpha \leq \beta^\sigma$.

In this analysis of coordination, a phrase 'X and Y' will always have the structure 'X [and Y]'. This is desired. The reason for this is that whether or not we assume G-access, the argument structure of Y will be the last element of the argument structure of and that matches.

We have ignored tense in this analysis. Notice however that the present analysis (without or without taking into account the tenses) does not produce the proper reading for the phrase. For it is clear that the phrase `venit et vidit` just as in the English equivalent (`He came and he saw.`) and is not a logical functor. For what the phrase says is that there was an event of coming and there was another event of seeing. Moreover, there is a natural expectation that the second event is after the first, so that and can be substituted by `and then`. We shall briefly return to the question of a natural interpretation for these things below. The present approach does also not capture those logical connectors that have two parts, such as `neither ... nor`, or even `if ... then`. To account for them, a far more complex semantic structuring has to be assumed.

There is an additional meaning of and that is often not distinguished properly from the logical meaning. This is the group forming meaning of and. The phrase `John and Mary` does not denote a conjunction in any sense of the word, at least if we wish to maintain the view that `John` and `Mary` denote individuals. Rather, and this will be the line that we shall take here, `John and Mary` denotes a group, consisting of both John and Mary. The group forming and is syntactically far more restricted than the logical one. It takes two things of the same kind and forms a group. We shall confine ourselves here to the use where it takes two individuals and forms a group. Its argument structure is the following:

(2.150)

$$
\begin{array}{c}
\text{/and/} \\
\begin{array}{|l|}
\hline
\langle x : \triangle\ \bigcirc : \begin{bmatrix} \textsc{num}: pl \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle y : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle z : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle. \\
\hline
x \\
\hline
x \doteq \{y, z\} \\
\hline
\end{array}
\end{array}
$$

The uses where it combines an individual and a group or a group and a group are as follows:

(2.151)

$$
\begin{array}{cc}
\text{/and/} & \text{/and/} \\
\begin{array}{|l|}
\hline
\langle x : \triangle\ \bigcirc : \begin{bmatrix} \textsc{num}: pl \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle y : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle z : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: pl \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\hline
x \\
\hline
x \doteq \{y\} \cup z \\
\hline
\end{array}
&
\begin{array}{|l|}
\hline
\langle x : \triangle\ \bigcirc : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle y : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle z : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle. \\
\hline
x \\
\hline
x \doteq y \cup \{z\} \\
\hline
\end{array}
\end{array}
$$

$$
\begin{array}{c}
\text{/and/} \\
\begin{array}{|l|}
\hline
\langle x : \triangle\ \bigcirc : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle y : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\langle z : \triangle\ \olessthan : \begin{bmatrix} \textsc{num}: sg \\ \textsc{cat}: ob \end{bmatrix} \rangle, \\
\hline
x \\
\hline
x \doteq y \cup z \\
\hline
\end{array}
\end{array}
$$

Obviously, in a language which distinguishes also a dual from a plural there are many more individual cases to be distinguished. They can be integrated into a single meaning, but it is worthwile pointing out that our present approach explains the fact that when and is used in the group forming sense the agreement in number (and gender/class and other features) are determined by certain rules taking into account the features of both NPs, while the logical and requires them to be the same and outputs the same argument structure. So, two verbs with singular agreement coordinated by logical and still take a singular subject, while two singular subjects coordinated by group forming and trigger plural agreement on the verb! It is actually no accident that x appears in the head section of the DRSs. Group forming and may in fact not only be used to coordinate two DPs that denote individuals or groups but can be used with events, places and many other things. (It will follow from the analysis of Chapter 4.4 that the individual or group is formed at the moment the DP is complete, and this will take care of the restriction that group forming and can coordinate only DPs and yields a group.)

What is needed to properly implement the above proposal is to implement a distinction between individuals and groups. Although from a purely ontological point of view groups are individuals (witness the fact that a subject of the kind `a group of tourist` triggers singular agreement), syntax operates on things differently depending on whether it analyzes them as individuals or groups. This means that one and the same thing may at one point be considered an individual and at the next moment a group. As this makes little difference in the actual semantics for the reasons discussed, we shall take it that there exists a feature which decides whether or not something is a group. We may actually take this feature to be PL versus SG. So, if a referent carries the feature PL it will act group like and if it carries the feature SG it will act individual like. (This points to the way in which the four meanings of and can be unified.)

Notes on this section. We shall return to the analysis of quantification and numerals in Chapter 4. The use of variables for argument structures is necessary in order to account for type polymorphism. However, there does not seem to be a need to use variables for the purpose, it would be enough to indicate that the AIS imports the entire argument structure. However, the use of variables makes matters clearer. The use of variables for DRSs is not strictly needed, as I have indicated above. If we replaced every possible DRS by an equation $p \doteq \delta$ then the handling of negation and disjunction is actually possible. The disadvantage of the latter approach is that uses too much notation for the effect that it achieves. I have chosen not to explore that variant.

Chapter 3

Argument Structure

This chapter shows in detail how agreement morphology interacts with the argument structure defined in the previous chapter. Morphology fills up the places in the argument structure that have been left blank by the lexicon. This allows to incorporate not only subject agreement but any kind of agreement with an argument and in any category (for example person, gender, number, case). We will show that while the number of possibilities is high so that languages only exhibit a fraction of them, for any kind of dependency there is a language where agreement is sensitive to it.

3.1 Morphosyntactic Representations

In his book [3], Anderson presents a theory of morphology and inflection. The main claim is that words are not segmentable into morphemes, or at least not in such a way that syntactic processes may operate on the words as strings. According to Anderson, morphological processes build up a morphosyntactic representation (MSR) in tandem with the word, and the morphosyntactic representation is the only thing that is visible to syntax. In this way, morphology becomes the interface between phonology and syntax. For the morphological rules change the word qua phonological string in tandem with the MSR.

Now what are these MSRs? Anderson assumes that they contain not only information about what the category of the word is but also what the categories of the subcategorized elements are. Moreover, specific properties of the word as well as its selected complements may be listed. Here is an example.

$$\begin{bmatrix} -\mathrm{V} \\ +\mathrm{Anim} \\ +\mathrm{Pl} \end{bmatrix}$$

This structure means that the word is a verb, has an animate plural actant. The MSRs are also layered. For example, the following is a representation of a noun with a possessor marking.

$$\begin{bmatrix} +\mathrm{Noun} & \begin{matrix} -\mathrm{me} \\ -\mathrm{you} \\ +\mathrm{Pl} \\ +\mathrm{Anim} \end{matrix} & \begin{bmatrix} -\mathrm{me} \\ -\mathrm{you} \\ -\mathrm{Pl} \\ -\mathrm{Anim} \end{bmatrix} \end{bmatrix}$$

This means the following. The word is of category noun and it is possessed by a third person plural animate. Itself, it is of third person, it is singular and inanimate.

A particular feature of Anderson's model is that the MSRs consist of elements whose meaning is formal and which may change from language to language. The structure of the representation also depends on the morphology. For even if—as Anderson claims—there are no morphemes, the representation must be built up. Anderson assumes that it is built up inside out and cumulatively. That is to say, lexical items have a rudimentary representation that gets filled more and more by the morphological processes. Features, once assigned, may not be overwritten. It follows that the morphological processes must follow in the way dictated by the MSR. The most interesting conclusion that can be drawn is that if there are agreement suffixes to a verb, their order is reflected in the MSR, and consequently in the syntax. The MSR of a verb is as follows.

1. Transitive verb: $[\mathrm{Tense}, \ldots, [F_1[F_2]]$
2. Intransitive verb: $[\mathrm{Tense}, \ldots, F_1]$

Here the outermost layer consists of the verbal categories (tense, aspect, mood, polarity), while F_1 contains the subject related features, and F_2 the object related features. It is vital to observe that F_1 and F_2 are merely distinguished by their relative position in the sequence. There are no special subject features or object features, as is commonly assumed in the Minimalist Program. Anderson then states the following principle.

> **Layering:** When a rule assigns features from a paradigmatic dimension D to a MSR $\mathcal{R}$ that already contains values on D, the original values are treated as a list item hierarchically subordinate to the new values.

It is not entirely clear what that means; presumably, agreement with respect to different dimensions are independent of each other. The number marking, for example, can be independent of the class marking. An illustration is given by Georgian verbs. The suffix `-t` means according to Anderson nothing but *has a plural actant*. Therefore, `-t` is added regardless whether the subject is in the plural or the object; it is not added if a more specific suffix (implying the plurality of some actant) is present. This is a familiar case of 'blocking'. We see therefore that the number marking in Georgian enjoys some independence of the person marking, the principles of which will be illustrated below.

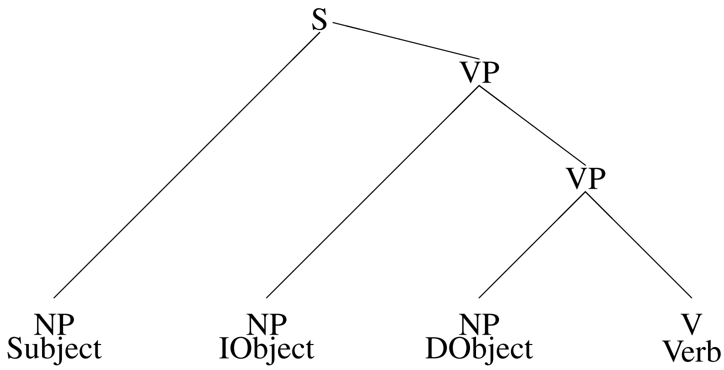
However, within in each dimension there is no choice as to the order in which the MSR must be filled, namely inside out. The relation with syntax is spelled out in the following rule.

> Agreement of a given item with elements in more than one position proceeds **cyclically**, with the structurally innermost elements triggering agreement before those that are structurally less close to the agreement item.

The structure of a basic sentence is depicted in Figure 3.1. We may interpret this as follows. When inserted into a syntactic structure, the verb may discharge its features ('check them', to use the modern terminology) provided that it finds an argument with matching features. The order in which this discharge proceeds is also inside out. The direct object has immediate access, then follows the indirect object, and at last the subject. Anderson proceeds somewhat differently, though, by putting down explicit rules for the connection between syntactic positions and positions in the MSR.

Anderson provides as major evidence for this architecture explicit morphological rules for Georgian verbs and for Potawatomi verbs. We will not discuss the

Figure 3.1: The Serialization of Verbal Arguments



details of this analysis here. We will merely use the Georgian example to illustrate how the MSR is built up. It is assumed that Georgian verbs carry a structure of the following form:

$$[\text{Tense}, \ldots, [-[-[-]]]]$$

The three slots in to the right correspond to the three arguments of the verb. Some verbs may actually not allow for certain arguments. In this case, some slot may be filled by the element ∅. It is by no means obligatory that the indirect object slot is filled by ∅ before any other slot may be filled. The following list of lexical MSRs may illustrate that point.

Class	Subcategorization	Lexical MSR
I	[S DO __]	[−[−]]
I+IO	[S IO DO__]	[−[−[−]]]
II	[e DO__]	[−]
III	[S __]	[−[∅]]
III+IO	[S IO __]	[−[−[∅]]]

A verb of a particular class has the MSR associated with that class. Its agreement properties must still be determined. Therefore, the following rules are given, which must be applied in strict succession.

1. (Direct Object Agreement, Obligatory) Copy the features and referential

index from a Direct Object NP to the Verb if present; if there is no Direct Object, add ∅.

2. (Indirect Object Agreement, Optional) Copy the features and referential index from an Indirect Object NP to the Verb if one is present; otherwise add ∅.

3. (Subject Agreement, Optional) Copy the features and referential index of a Subject NP to the Verb if one is present; otherwise add ∅.

4. (Dummy Agreement) Add ∅.

The model of Anderson is very interesting for our purposes, since it gives a detailed account of the morphological architecture of verbs and how the representations may be built up. There are, however, several questions concerning this approach. First of all, the relation between morphology and syntax is less clear than Anderson thinks. Postulating agreement rules of the kind above principally amounts to giving up a tight connection between the slots of a MSR and the grammatical function of the actant. We must however assume that the grammatical function is directly determined by the position in the MSR. In an intransitive verb there is only a subject. A transitive verb only has a subject and a direct object, and ternary verbs have subject, indirect object, object, in that order. We will not assume that grammatical functions are directly related with cases. One exception may be the subject, which in the overwhelming majority of cases is nominative. Otherwise, there exist dative direct objects in German (see [42]), and in Georgian the case assignment is determined by a combination of several factors, including class of the verb and its tense. (See also [56].)

In what is to follow we will outline a somewhat different model. The main difference is that while Anderson is interested in the morphology of words, we are interested in the connection between MSRs, syntax and semantics. Mainly, what is Anderson's MSR is a reduced part of what we call argument structure. The list structure of the MSR is a consequence of the ordering in the argument structure. We agree with Anderson that inflection is the result of a process that enriches the MSR, but we also claim that it additionally enriches the semantics of a word to determine its syntactic behaviour. Rather than viewing the word as a member of a syntactic structure which determines the agreement features (by copying), we think that the agreement features are present prior to the building of the syntactic structure. For example, if the verb does not agree with the subject, no syntactic structure can be built, and hence the sentence is ungrammatical. An additional difference with Anderson is that we believe that the dimensions of items have a universal order, and this order can be motivated from a cognitive-semantic point of view. A last note on morphology is in order. Pace Anderson, we shall use sequences of morphemes to denote lexemes. Since these sequences are regimented by the argument structures in their linear order, we shall commit ourselves to the view that there exist elements that can be lined up in this way. So, for example, we shall assume that Latin hominis (*of the man*) is a sequence consisting of roughly the elements homo (*(the) man*), SG (singular) and GEN (genitive). [1] Thus, in writing the sequence

(3.1) $\texttt{homo}^\frown\textsc{sg}^\frown\textsc{gen}$

appears as hominis. Moreover, roots are marked by a superscript $-^{\checkmark}$, so that no uncertainty arises whether the element is taken as inflected or as root. So we get

(3.2) $\texttt{hominis} = \texttt{homo}^{\checkmark\frown}\textsc{sg}^\frown\textsc{gen}$

However, we shall not assume that the elements are morphemes. Rather, the best term in this case is **inflectemes**. The reason is that their segmentation is not justified by a segmentability of the string itself into three distinct parts which follow each other. Rather, we observe that in order to arrive at the fully inflected word, two pieces of meaning must be added. The overt reflex of each individual piece can often hardly be seen, but they tend to occur in bundles that can be spelled out each as whole. (See Section 3.7 for a discussion.) For example, in the Latin verbal inflection there is a tense morpheme, but no person morpheme. In sequel, we will suppress such detail and speak of morphemes, but the reader is asked to keep in mind that the reality is more complex than that.

[1] We shall use sans serife whenever the string appears more or less as it is in print. However, we shall use small caps in order to quote a unit. Thus gen would be the sequence which appears in print as 'gen', but GEN might actually appear in many forms. This allows us to keep away from problems of allomorphy for example. But it also allows us to use concise notation to denote the entity that we mean rather than having to use some string whose meaning in the given context would be rather unclear. It is much more concise to use the label GEN than to use any of its forms, say -e, to give one example.

3.2 The Role of Inflection

If a lexical item subcategorises for another item (for example, verbs subcategorizing for nouns) then it can—but need not—show what is generally called *agreement* with that item. Agreement means that the form of the verb, say, in some way reflects the property of the noun that fills the subcategorisation slot. A very typical case is subject-verb agreement. The verb agrees in all or some of the properties that the subject has. We give an example from German. The following sentences are grammatical.

(3.3)

`Ich gehe.` *I walk.*	`Wir gehen.` *We walk.*
`Du gehst.` *You walk.*	`Ihr geht.` *You walk.*
`Er/Sie/Es geht.` *He/She/It walks.*	`Sie gehen.` *They walk.*

The verb forms may not be interchanged, so *`du geht` is ungrammatical. Hence we say that the verb *agrees* with the subject in number and person. (In some languages, agreement in gender or class and definiteness also exists.)

The phenomenon of agreement is pervasive. However, languages differ very much with respect to the richness of the agreement systems. There is a continuum of possibilities. We have what is known as noun incorporation languages (Mohawk, Lakhota, Icelandic). They present the richest form of agreement system. (However, as will become apparent from the discussions in this chapter, noun incorporation is not necessarily a form of agreement.) Then come the purely morphological types, starting with languages having double or threefold verbal agreement (Basque, Georgian, Mordvin), to languages with typically unary agreement only (German, Latin). At the bottom end are languages which have next to no inflection (English) and those without inflection at all (Chinese). It is therefore not appropriate to introduce a dichotomy between languages that show agreement and those languages that do not. However, there is a rather intriguing question, namely whether languages without agreement can in fact be seen as languages that have agreement like any other language, only that the name space is rather poor. To give some more examples, the category of *gender* is morphologically unmarked in Hungarian. Neither verbs nor adjectives agree in any sense of the word with the noun in gender. We can now either say that Hungarian has underlying categorization into gender but overtly there is no distinction in gender; or we can say that Hungarian simply has no gender. The latter is technically equivalent to saying that Hungarian has only one (syntactic) gender and that all gender distinctions are purely semantic. For example, there is a word for 'woman' and 'man'. The distinction between the two is semantic in Hungarian as in any other language. For knowing what these words mean is to know distinguish gender with humans. But this would be all there is to Hungarian gender in language, if the second view is adopted. There is every reason to prefer the second over the first. To see what sort of difficulties we face, let us assume that a language like Hungarian has several genders. Then the obvious questions is: how many and which genders are there? And how is the world classified into genders? Obviously, if we do not see any overt classification in the morphology or syntax, and since we are not assuming that gender is semantic we must assume that there must be some a priori way to classify things into classes or genders. However, if we look around to other languages, we see that this view is against all odds. Gender systems of the languages in the world are in fact quite incompatible with each other (see [22] for a rather exhaustive exploration of gender and its function in language). In Indo-European we have three (German, Latin, Greek, Rumanian) or two (French, Spanish) or one (English). These are *masculine*, *feminine* and *neuter*. The threefold system, however, does not apply in a semantic manner. These labels apply in all of these languages more or less only to living things in a uniform way. To other objects they are assigned arbitrarily. The sun is feminine in German (not neuter, as in English). It is masculine in French, while the moon is feminine in French but masculine in German (and, again, not neuter). Other languages have up to twenty genders (or classes), and the division of things into these classes is likewise filled with arbitrary classifications. There is no natural way in which the things that we name by words fall into classes out of which the genders in the various languages of the world are derived. This is not to say that there are no languages in which the gender assignment is semantic. Dyirbal (Australia) and Ojibwa (Algonquian) are such languages. Here, as in many other languages as well, the gender assignment is part of the (often mythical) world-view.

Of course it is absurd to say that the languages without gender of class distinctions lack the means of expressing them. Naturally, the notions of 'masculine', 'feminine', 'animate' and so on exist in all languages. The question is whether they are syntactically relevant. Moreover, as we will see later, there is an additional question as for the distinction between syntactic relevance and morphological relevance. Agreement cares only about the latter, by definition, but nevertheless we will argue that the name space consists of all those distinctions that are syntactically relevant.

The basic claim we are advancing is the following. Languages generally distinguish events and objects. Furthermore, object arguments may be classified by means of case, gender (class), number, person, and definiteness. Event arguments may also be classified, but the classificatory system of events is less clear than that of objects. Any lexical item subcategorizing for some other lexical item may show agreement in some (or all) of the dimensions of the name space. Usually, there is some syncretism, but we assume simply that the limit of agreement is set by the number of arguments, their type and the name space.

Restriction 4 *The inflectional morphology provides distinctive forms for a lexeme or morpheme only up the distinctions that can be made with respect to the identifying names of the referents in the argument structure.*

For example, HOMO in Latin forms a paradigm with eight members.

(3.4) $$\textsc{homo} = \{\texttt{homo}, \texttt{hominis}, \texttt{homini}, \texttt{hominem}, \texttt{homine}, \texttt{homines}, \texttt{hominum}, \texttt{hominibus}\}$$

The paradigm itself may be associated with the following semantic structure

(3.5) $$\begin{array}{|l|} \hline \langle x : \vartriangle : \left[\begin{array}{lll} \textsc{cat} & : & ob \\ \textsc{case} & : & \top \\ \textsc{num} & : & \top \\ \textsc{class} & : & masc \end{array}\right] \rangle \\ \hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}$$

We see that the gender is fixed, number and case are not. Any particular member of the paradigm corresponds to a representation which differs from 3.5 only in that the places of number and case have been filled by specific values. (So, a member of the paradigm has an AVS that is a name. The paradigm is thus abstract.) For example, `hominem` becomes

(3.6)

$$\begin{array}{|l|}\hline \texttt{/hominem/} \\ \hline \langle x : \triangle : \begin{bmatrix} \textsc{cat} & : & ob \\ \textsc{case} & : & acc \\ \textsc{num} & : & sg \\ \textsc{class} & : & masc \end{bmatrix} \rangle \\ \hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}$$

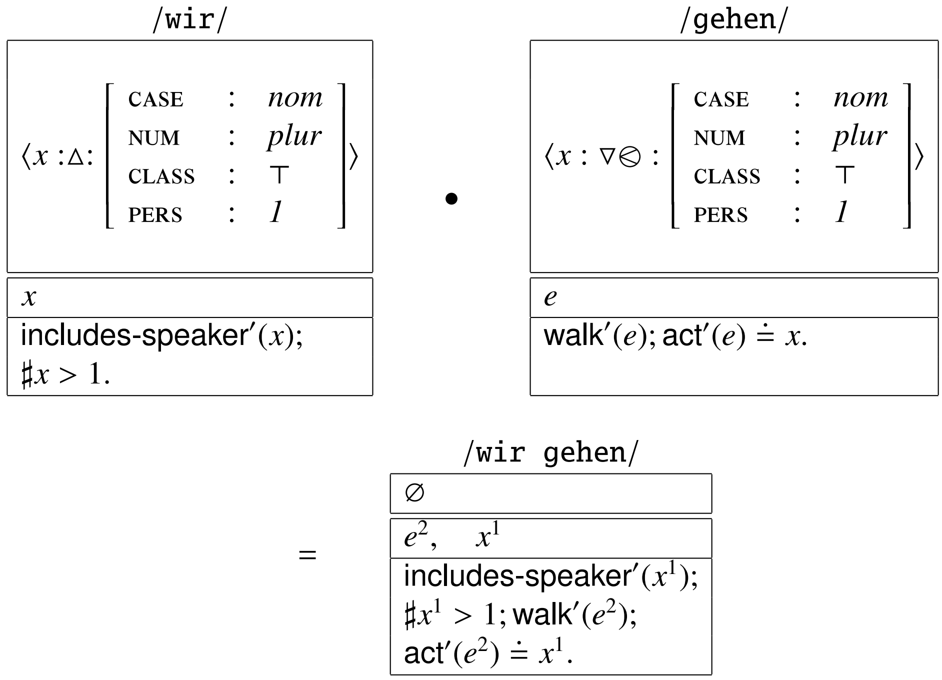
The paradigm should not be confused with the *stem*. The stem is a lexical entry. It looks like this:

(3.7)

$$\begin{array}{|l|}\hline \texttt{/homin}^{\checkmark}\texttt{/} \\ \hline \langle x : \triangle : \begin{bmatrix} \textsc{cat} & : & ob \\ \textsc{case} & : & \star \\ \textsc{num} & : & \star \\ \textsc{class} & : & masc \end{bmatrix} \rangle \\ \hline \varnothing \\ \hline \mathsf{man}'(x) \\ \hline \end{array}$$

Here, a $\star$ is put where the value is 'undefined' rather than underspecified. Inflection and derivation can change $\star$ into some other value, of course. Likewise, verbs and adjectives form paradigms, which are unfortunately rather large. In the case of an adjective, all three nominal dimensions are free. The forms of the adjective vary in all three dimensions with the name of the referent. In the case of a verb, the paradigm provides only the cases of the arguments. Hence, the form may vary with all arguments along gender, number (and person). However, distinctions are morphologically visible only for number and person of the noun phrase of the subject (which is the nominative marked noun phrase), and furthermore tense and aspect. Restriction 4 does not limit the number of possible overt forms, since cases of allomorphy are not included here. So, in a way this principle is vacuous if it would only be used to tell us how many distinct forms we should expect. Rather, it tells us what counts as inflection and what as allomorphy.

Figure 3.2: Inflection and Merge



The rationale behind Restriction 4 is as follows. In syntax there are various means of identifying an argument. One is its position (left or right) and the other is its form. To repeat the German example, the verb and the subject can only merge if they agree in the name of the subject. This is shown in the case of the sentence `Wir gehen` ('we walk') in Figure 3.2. Notice that neither the subject nor the verb indicate the gender of the actor. The various possibilities of instantiating these names can then (but need not be) morphologically reflected by different forms. This would predict among other that a verb can agree with as many arguments as it gives identifying names for, hence up to three arguments (since four arguments are extremely rare). This is indeed the case, in Basque, for example. So this thesis is interesting because it tells us what morphological complexity we should expect cross-linguistically. It rarely so happens that languages exhibit all possible agreement distinctions that can be drawn in this way, but we will show that almost any phenomenon not excluded by the above restriction is actually instantiated in some language. Moreover, it seems that the restriction is actually vacuous, since if there is no referent, how can agreement at all be possible? But we will see that there are some very difficult morphological facts that seem to violate that principle. It does indeed have a nontrivial kernel, which we are going to isolate.

In the next sections we will give ample evidence for our claim that the name space together with the subcategorised arguments provide the basis for inflection. Before we can do so, we need to spend a little time explaining the structure of the name space. Almost all agreement is nominal in character, so we need to worry mainly about the space of nominal referents. Here we find mainly four dimensions: gender (class), number, person and case. Some languages also distinguish definite and indefinite. However, not all languages allow for the factorization along these dimensions, and in at least one instance it is also not so straightforward. The problematic case is the factorization of person and number. If one is not careful enough in defining the semantic contribution of person, number and person become unnecessarily intertwined. For example, if we say that first-person means 'consists of speakers only', then first person plural would be reserved for cases where there is a multitude of speakers. However, this is not what 1st plural means. It means only that the group of speakers is included. So, let us therefore see how a proper definition might go.

Virtually all languages distinguish three persons, which function roughly as in English.[2] Let us call them therefore 1, 2, and 3. The numbers stand for speaker (1), addressee (2) and none of the previous (3). The canonical situation (but by no means the only one) is that there is a single speaker and a single hearer. If we have a (nonempty) set P of people, several possibilities occur. (a) P is a singleton set. Then the unique member is the speaker (1), the hearer (2) or some third person (3). (b) P has more than one member. Then the members can be a combination of the following persons: 3, 1+2, 1+3, 2+3 and 1+2+3. The so-called 1st plural in Indo-European languages lumps together 1+2, 1+3 and 1+2+3, while the 2nd plural is simply 2+3 and the third plural is 3. There are however languages which distinguish a so-called inclusive 1st plural from an exclusive 1st plural. (Moreover, Tok Pisin distinguishes an exclusive dual and an inclusive dual, see [34], page 67. In Table 3.1 taken from [23] we show the pronominal system of a language that has a dual and a trial and distinguishes inclusive from exclusive.) The inclusive 1st plural lumps together 1+2 and 1+2+3, while the exclusive 1st plural is the combination 1+3. Now there are two basic conceptions. The first is to say that to be *1st* means: consists of speaker(s), to be *2nd* means: consists of hearer(s),

[2]For exceptions see [87].

and to be *3rd* means: consists of non-speakers and non-hearers. However, this is very problematic, since the mixed groups are quite frequent. For then, logically speaking, none of the combinations 1+2, 1+3, 1+2+3 is the real plural of the 1st person.

The following analysis into distinctive features is applicable almost exceptionless across languages. With respect to person, there are two features: *includes speaker* (INCL:1) and *includes hearer* (INCL:2). We denote by EXCL:1 the negation of INCL:1 and by EXCL:2 the negation of INCL:2. These two features produce four distinctions:

	INCL:2	EXCL:2
INCL:1	1.INCL	1.EXCL
EXCL:1	2	3

Notice that there is no feature for the third person: it is defined only negatively. This leads to three possibilities in the singular, since a single individual cannot both be speaker and hearer. In the dual, trial and plural we get 4 possibilities. For a language with singular, dual, trial and plural there is a total of 15 combinations of person and number, but the distinctive features can separate only 11. The language of 3.1 shows indeed all 11 possibilities.

There are also languages in which there is no clear division between number and gender, at least if we follow the traditional usage of these terms in the established philologies. Such are the Bantu languages. For example see [102] for Zulu (and below) and [38] for Bemba. In these languages, some classes are singular and some are plural. For example, the class 1 is singular, while class 2 is plural. A set of several class 1 objects is class 2. However, we will say that class 2 is simply the plural of class 1.[3] Therefore, we form the archiclass 1-2, members of which are either of class 1, when singular, or of class 2, when plural. This is how we will view the matters here, although we will stay with the conventional terminology, for ease of comparison. Some Bantu languages are actually more complex than that. They even do not have neat division into singular classes and plural classes. The same class can be a singular for certain nouns and a plural for others ([**?**]).

We conclude this section with some terminological notes on grammatical relations. The main verbal arguments are distinguished by three **grammatical re-**

[3] It is from a morphological point of view not possible to factor the classes into a combination of class and number.

Table 3.1: Personal Pronouns of a Language spoken on the Annaton Island (Melanesia)

ainjak	1.SG	aiek	2.SG
aiumrau	1.DU.EXCL	aijaurau	2.DU
akaijau	1.DU.INCL		
ajumtai	1.TR.EXCL	aijautaij	2.TR
akataij	1.TR.INCL		
aijama	1.PL.EXCL	aijaua	2.PL
akaija	1.PL.INC		

lations (**GR**s), which are simply numbers, 1, 2 and 3, corresponding to subject, object and indirect object (see Section 6.2. In other traditions, one speaks of *grammatical functions* (**GFs**) and these are **subject** (S), **actor** (A) and **undergoer** (U). *Subject* is reserved for the subject of an intransitive verb, *actor* for the subject of an transitive verb, and *undergoer* for the object of an transitive verb. The *nominative* case is assigned to those NPs which have the GFs of subject and actor, the absolutive is assigned to those NPs with GFs subject or undergoer. The accusative is reserved for the undergoer, and the ergative for the actor. This is the general scheme. However, matters can get quite complex, as we will see below. A verb therefore determines the case only of its third complement, which may be a beneficiary, a locative, for example (see [20]).

The case names are defined as follows. The **nominative** comprises S and A, the **accusative** only U. The **ergative** comprises S and U, the **absolutive** only A. Some linguists have also postulated that there are basically three cases, A, S and U and that the other cases are derived from them. In fact, Foley reports ([34], page 105) that the Papuan languages Yimas and Anggor distinguish by cases all three functions, which gives support to this idea.

3.3 The Nominal Group

We have argued that lexical elements can show an overt reflex for all properties of arguments that they subcategorise for. In many cases this will allow for rather rich morphological paradigms. Although we know of no language that displays all possible agreement patterns, we will show by way of examples that any of the predicted possibilities is attested in some language.

Before we discuss the various paradigms, let us state what the dimensions are along which objects are classified. These are *person*, *class*, *number*, *definiteness* and *case*. It is exactly in the order just given that they can be added to a stem. Clearly, case denotes a relation with the verb (see the discussion in Chapter 5), and it is a relation with an object that is provided by the noun phrase. So, the case is the outermost layer, and it corresponds in syntactic terms with the PP. The object is singled out by a description, and this description is provided by the NP. In between the NP and the PP is the DP. At the DP level it is specified whether or not the description picks out a unique or salient object. This is marked by the definiteness marker or by the article. So the NP describes an object. If the object is first of second person, then it is already determined by the context. Class information is redundant, but may of course be present. Number marking, however, may be necessary, because there are cases where person marking does not suffice. For third person objects, the class information becomes relevant. The number is actually independent of the class. However, the typical group is a homogeneous group of objects described by some noun, and each object therefore has a class, namely that given by the noun. This indicates why number marking is outside of class.

These facts motivate the following conclusion. There is a hierarchy of dimensions, and if some dimension is filled, then the dimension of the lower rank may not be filled any more. Since all dimensions must be filled, the inflection proceeds from lower to higher rank. Moreover, full syntactic categories that are provided with some features must have been provided with all features of lower rank as well. Now, the nominal categories may be distinguished according to the features that have been assigned a value.

<table>
<tr><td>Person</td><td>Class</td><td>Number</td><td>Definiteness</td><td>Case</td></tr>
<tr><td colspan="3">NP</td><td></td><td></td></tr>
<tr><td colspan="4">DP</td><td></td></tr>
<tr><td colspan="5">PP</td></tr>
</table>

	Person	Class	Number	Definiteness	Case
Pronoun	–	–	–	–	–
Adjective	+	–	–	–	–
Noun	+	+	–	–	–
Numerals	+	–	+	–	–
Determiner	+	–	–	+	–

These facts will motivate the layering of an NP in the way described above. Notice that almost all elements are specified for person. By default it is 3rd person.

The semantics of case will be treated extensively in Chapter 5. For the remaining dimensions, the facts are somewhat less complicated. We generally assume that in each dimension there is an element ⋆ that stands roughly speaking for 'undefined'. Thus, in contrast to Anderson, each dimension is actually present in the argument structure, only that initially many features carry the value ⋆. The need for this is the following. By the logic of feature structures, if a feature is not instantiated, then it has any value—but there is no notion of undefinedness. This is therefore introduced by this element. However, it is not appropriate to think of this element as denoting an undefined value. (This will become particularly clear when we discuss the case-fan of languages.) For ⋆ must not be confused with having no name at all, which by the logic of merge leads to a failure of identification. We shall therefore say that the ⋆ denotes a special generic value. In some cases may be the only one the language has. For example, the gender feature in Hungarian may have only the value ⋆. If an element carries [FEAT : ⋆], this does not mean that it is ill-formed or illegitimate. We assume for example that Hungarian adjectives simply do not inflect. Their argument structure is therefore maximally undefined (in the sense that all features receive value ⋆). Number is assigned to the entire phrase, and not to the adjective. We shall say that it is the specific task of morphology to identify those feature structures that are legitimate, this is to say those that correspond to an independent unit. For example, morphology is responsible for saying that in Hungarian the adjective is pronounceable as it is, without any change, while nouns and determiners need case (that is, may not have ⋆ assigned to the case-feature). However, nouns and determiners are drawn from the lexicon with the case-feature instantiated to ⋆, which means that morphology must add a case-morpheme. In this way, morphology also takes care (at least partly) of the obligatoriness of marking.

The layering of the dimensions can be achieved by a particular choice of argument structure. The argument structure of 1ST will be as follows (assuming that

Figure 3.3: Relational (right) and Nonrelational Nouns (left)

/Haus$^{\checkmark}$/

$$\langle x : \vartriangle : \left[\begin{array}{lll} \textsc{class} & : & \mathit{neut} \\ \textsc{number} & : & \star \\ \textsc{case} & : & \star \end{array}\right]\rangle$$

$\varnothing$

house$'(x)$.

/Vater$^{\checkmark}$/

$$\langle x : \vartriangle : \left[\begin{array}{lll} \textsc{class} & : & \mathit{masc} \\ \textsc{number} & : & \star \\ \textsc{case} & : & \star \end{array}\right]\rangle$$

$$\langle y : \triangledown : \left[\begin{array}{lll} \textsc{class} & : & \top \\ \textsc{number} & : & \top \\ \textsc{case} & : & \mathit{gen} \end{array}\right]\rangle$$

$\varnothing$

father$'(x, y)$

the suffix is attached through fusion and not merge).

(3.8) $\langle x : \blacklozenge\oslash : [\textsc{pers} : \star \mapsto \mathit{1}]\rangle$

(The directionality is specified for a suffix. 1st could of course be a prefix or both.) The particular requirement on the argument structure is that all dimensions be undefined. The argument structure of a class suffix fem is as follows:

(3.9) $$\langle x : \blacklozenge\oslash : \left[\begin{array}{lll} \textsc{pers} & : & \top \\ \textsc{class} & : & \star \mapsto \mathit{fem} \end{array}\right]\rangle$$

Here $\top$ only means that the value is everything except $\star$. Hence plural can be assigned only if person is assigned already. Likewise, we may arrange the argument structure of the other dimensions. However, it does seem to us that this way of bringing about the facts is not very helpful since it offers no insight into them.

Nouns Nouns are characterized by the fact that their argument structure contains $\langle x : \vartriangle \bigcirc : A\rangle$, where x is an object. Nouns are object arguments. Nouns typically have only one referent. Some nouns also take other arguments, like `destruction`, but these arguments tend to be less obligatory than those of verbs. It is therefore observed that nouns show agreement only with this one argument, and very frequently they do. Examples are Latin, Ancient Greek and Finnish (number, case) and English (number). Nouns in Hungarian show only optional agreement in number. We give a German example, `Haus` (‘house’).

		Number→	
		SG	PL
Case	NOM	`Haus`	`Häuser`
↓	GEN	`Hauses`	`Häuser`
	DAT	`Hause`	`Häusern`
	ACC	`Haus`	`Häuser`

So, we have the following equation

(3.10) $\texttt{Haus}^{\checkmark}{}^\frown\textsc{pl}{}^\frown\textsc{dat} = \texttt{Häusern}$

In German, the dimensions of agreement are class (gender), case, and number. In addition, adjectives show a three way agreement with respect to the type of determiner, see below. In Figure 3.3 we have shown the lexical entry of an nonrelational and a relational German noun. Agreement is only with the first referent, and it is with respect to case and number. The class is fixed. Relational nouns select genitive case for their complements. Since they do not show any agreement, we have assumed that the value of CLASS and NUMBER is $\top$, which means anything other than $\star$. (So, $\top$ represents the disjunction of all values different from $\star$.)

Pure head marking languages have no case, they simply show number, person and/or class-agreement on the verb. Hence in such languages, typically only number agreement remains. Such a case are Bantu languages. We give an example from Zulu (see [102]). Table 3.2 lists the prefixes for nouns of the various classes. It can be seen that there is no clear factorization of the class system into class and number as in Indo-European languages. The classes are either singular (1, 1a, 3, 5, 7, 9, 11) or plural (2, 2a, 4, 6, 8, 10) or neutral (14 and 15). Each singular class has an associated plural class; the plural class of 11 is 10. Into the classes 14 and 15 typically fall nouns that generally have no plural (mass noun, liquids etc.).

Adjectives Adjectives are similar to nouns, only they are object adjuncts. The facts concerning adjectives are similar to those of nouns with the exception that the entry for adjectives is also underspecified with respect to gender. Adjectives will be studied in detail in Chapter 4. We give some examples from Latin. Most

Table 3.2: Zulu Noun Class Prefixes

Singular Prefixes		Plural Prefixes	
Class 1	um-/umu-	Class 2	aba-
Class 1a	u-	Class 2a	o-
Class 3	um-/umu-	Class 4	imi-
Class 5	il-/ili-	Class 6	ama-
Class 7	isi-	Class 8	izi-
Class 9	in-/im-	Class 10	izin-/izim-
Class 11	u-/ulu-		
Class 14	ubu-		
Class 15	uku-		

adjectives are nonrelational, for example `magnus` ('big'), but there is a small class of adjectives that require an additional complement, which is typically in the genitive case. Such an adjective is `avidus` ('greedy'). We give the paradigm for `magnus` ('big') in Table 3.3. As we will expect for a head marking language adjectives agree with the modified noun in class. This is indeed the case; the Zulu class agreement markers are displayed in Table 3.4. As notes Zemb [104], the adjectives can appear in at least four types of environments: as modifiers of nouns, as modifiers of adjectives, predicatively with typical predicative verbs ('to be') and predicatively with normal verbs. (See also [**?**].) The first environment is the classical case of adjectival function and needs no comment. The second function, modifier of an adjective, might be surprising. Typically, only adverbs can be put into this environment. This is the rule, but there exist modifiers of adjectives that cannot be used as adverbs, such as `very`, `extremely`. Thirdly, the predicative function is again a rather standard one for adjectives, while it is noticeable that many verbs tolerate an additional predicative adjective.

The question with respect to these four environments is which of them is filled by adjectives and which is filled by adverbs. Clearly, nominal modifiers are adjectives, and, typically, modifiers of adjectives are non-inflectible (which automatically makes them adverbials in many languages). We need not concern ourselves with these types. But languages differ with respect to the predicative function, whether it be filled by adjectives or by adverbs, or to be exact, whether adjectives are different morphologically depending on whether they are used predicatively

Figure 3.4: Relational (right) and Nonrelational Adjectives (left)

/magn$^{\checkmark}$/

$$\langle x : \Diamond\oplus : \begin{bmatrix} \text{CLASS} & : & \star \\ \text{NUM} & : & \star \\ \text{CASE} & : & \star \end{bmatrix} \rangle$$

$\varnothing$

big$'(x)$.

/cupid$^{\checkmark}$/

$$\langle x : \Diamond\oplus : \begin{bmatrix} \text{CLASS} & : & \star \\ \text{NUM} & : & \star \\ \text{CASE} & : & \star \end{bmatrix} \rangle$$

$$\langle y : \nabla\oplus : \begin{bmatrix} \text{CLASS} & : & \top \\ \text{NUM} & : & \top \\ \text{CASE} & : & gen \end{bmatrix} \rangle$$

$\varnothing$

greedy$'(x, y)$

Table 3.3: Latin Adjectives: magnus 'big'

			$\gamma \rightarrow$ MASC	FEM	NEUT
κ, ν $\downarrow$	SG	NOM	magnus	magna	magnum
		GEN	magni	magnae	magni
		DAT	magno	magnae	magno
		ACC	magnum	magnam	magnum
		ABL	magno	magna	magno
	PL	NOM	magni	magnae	magna
		GEN	magnorum	magnarum	magnorum
		DAT	magnis	magnis	magnis
		ACC	magnos	magnas	magna
		ABL	magnis	magnis	magnis

Table 3.4: Zulu Adjective Agreement Prefixes

CLASS	NOUN PREFIX	ADJECTIVE PREFIX
CLASS 1	um(u)-	om(u)-
CLASS 1A	u-	om(u)-
CLASS 2	aba-	aba-
CLASS 2A	o-	aba-
CLASS 3	um(u)-	om(u)-
CLASS 4	imi-	emi-
CLASS 5	il(i)-	eli-
CLASS 6	ama-	ama-
CLASS 7	isi-	esi-
CLASS 8	izi-	ezin-
CLASS 9	in-/im-	en-
CLASS 10	izin-/izim-	ezin-
CLASS 11	u(lu)-	olu-
CLASS 14	ubu-	obu-
CLASS 15	uku-	oku-

or attributively. There are languages in which this is the case. For example, in German, adjectives do not inflect when used predicatively. We take this as a diagnostic that they are (morphological) adverbs.

(3.11) `Der auf seine Erfindung stolze Hans`
the onto his invention proud-MASC.NOM.SG Hans
Hans, who was proud of his invention,...

(3.12) `Hans war stolz auf seine Erfindung.`
Hans was proud-∅ onto his invention
Hans was proud of his invention.

In English, however, the adjectives do not change to adverbs when used predicatively.

(3.13) `They were nice to us.`
*`They were nicely to us.`

(3.14) `This was nicely said.`
*`This was nice said.`

(3.15) `Maria was good to her mother.`
*`Maria was well to her mother.`

(3.16) `The car performed well.`
*`The car performed good.`

In French, the adjectives inflect normally when used in this environment.

(3.17) `La femme était heureuse.`
the woman was happy-FEM.SG

In some languages, the adjectives have a different form when used attributively and when used predicatively. Such an example is Zulu. The rule is the following. When an adjective is used predicatively, the initial vowel of the class prefix is dropped.

(3.18) `isitolo esi-khulu`
store 7:ATTR-big
big store

(3.19) `Isitolo si-khulu.`
store 7:PRED-big
The store is big.

(3.20) `umuntu om-dala`
person 1:ATTR-old
old person

(3.21) `Umuntu m-dala.`
person 1:PRED-old
The person is old.

Hence, predicative adjectives form a class of their own. In Sami (Lappish), the adjectives are inflected as nouns when used predicatively, similarly in Hungarian. When used attributively, the adjective is not inflected at all. (So this is the exact opposite situation as in German.) When used predicatively (or as a noun), it inflects for number and case. (See [70].)

(3.22) `Mánná lea čeahppi.`
child-NOM.SG is talented-NOM.SG

(3.23) `Mánát leat čeahpit.`
child-NOM.PL are talented-NOM.PL

(3.24) `čeahpes mánná`
talented child-NOM.SG

(3.25) `čeahpes mánát`
talented child-NOM.PL

In Votiac, the facts are still different. Votiac nouns distinguish an indefinite and definite form. When used attributively with a definite noun, adjectives inflect, otherwise not. Moreover, there are different forms for adjectives depending on whether they are used as nouns or whether they are used predicatively.

There is an additional property of adjectives that deserves mentioning. In some languages, adjectives have the choice to be on either side of the noun. However,

Table 3.5: The Georgian Adjective

	SG		PL	
NOM	maɣal-i	mta	maɣal-i	mt-eb-i
ERG	maɣal-ma	mta-m	maɣal-ma	mt-eb-ma
GEN	maɣal-i	mt-is	maɣal-i	mt-eb-is
DAT	maɣal	mta-s	maɣal	mt-eb-s
INST	maɣal-i	mt-it	maɣal-i	mt-eb-it
ADV	maɣal	mta-d	maɣal	mt-eb-ad
VOC	maɣal-o	mta-o	maɣal-o	mt-eb-o

	SG		PL	
NOM	mta	maɣal-i	mt-eb-i	maɣl-eb-i
ERG	mta-m(a)	maɣal-ma	mt-eb-ma	maɣl-eb-ma
GEN	mt-is(a)	maɣal-is(a)	mt-eb-is(a)	maɣl-eb-is(a)
DAT	mta-s(a)	maɣal-s(a)	mt-eb-s(a)	maɣl-eb-s(a)
INSTR	mt-it(a)	maɣal-it(a)	mt-eb-it(a)	maɣl-eb-it(a)
ADV	mta-d(a)	maɣal-ad(a)	mt-eb-ad(a)	maɣl-eb-ad(a)
VOC	mta-o	maɣal-o	mt-eb-o	maɣl-eb-o

they will inflect differently when preceding it than when following it. An example is Georgian shown in Table 3.5 (cf. Fähnrich [31]). As one can see, the endings of the postposed adjective are like those of a noun. In Modern Greek, the adjective must be repeated together with the article, and in Hungarian it must pick up the case. These facts suggest that the adjective that is postposed rather than preposed (the normal position in all these languages) becomes the head of a new noun phrase, which speaks about the same object as the previous noun phrase. Similar facts have led Hale [43] to propose a general schema for so-called non-configurational languages: elements that free themselves from the structure (in this case adjectives), must pick up some overt inflection. The following example from Hungarian may illustrate this. (Here, a may simply be treated as a left delimiter of a noun phrase. It has very little meaning.)

(3.26) Voltam ebben a házban a fehérben.
was this-INESS PRT house-INESS PRT white-INESS
I was in this house, the white one.

We explain these facts as follows. Adjectives can take an empty N, glossed as ONE, to form an NP that is postposed. The argument struatucre of the rsulting expression is that of a noun. This analysis will be worked out in Section 4.5. In Hungarian, adjectives may become nouns without further derivation. When they do so, they may in fact head their own NP, which allows them to appear in a different position than they normally would. Of course, they are then banned from their original position. In other languages—Latin belongs to this class—adjectives may simply appear on either side of the noun. They do not have to be transformed into nouns. The position they take is determined by other factors, for example stylistic ones.

(3.27) `Fuerunt in magno templo.`
be-PERF-3.PL in big-ABL.SG.MASC temple-ABL.SG

(3.28) `Fuerunt in templo magno.`
be-PERF-3.PL in big-ABL.SG.MASC temple-ABL.SG
They have been a big temple.

Determiners Determiners are syntactically outside of the NP, but we nevertheless regard them as nominal. A determiner will typically set the value of the definiteness dimension. A noun phrase is neither definite nor indefinite. It expresses a property of an object or a group. Only the determiner will tell us whether this group is definite or indefinite. Quantifiers, which usually are treated on a par with determiners, will be left out of discussion here. They have been discussed in Section 2.9. The semantics of definiteness is rather hard to tie down. Usually it means that the entity is contextually given or salient. This however is to a large extent a pragmatic category, for it helps to track the entity in the discourse, by telling the hearer that it has already been talked about etc. For these reasons, we will leave the semantics unspecified and regard determiners as transformers that reset the value of definiteness from undefined to definite or indefinite. Hence, determiners agree with the complement in gender, case, and number. This is so in all languages we have studied. An example in Latin is given in Table 3.6. There are languages in which the noun is inflected according to definiteness. An example is Mordvin. We show in Table 3.7 only part of the case system of Mordvin (data from [55]). It can be seen that the nouns inflect differently according to whether the noun is indefinite (indeterminative inflection) or definite (determinative inflection). In Rumanian, definiteness is indicated by an affix to the noun.

Figure 3.5: Argument Structure of Determiners

$$\langle x : \Diamond\circleddash : \begin{bmatrix} \text{CLASS} & : & \star \\ \text{NUM} & : & \star \\ \text{CASE} & : & \star \\ \text{DEF} & : & \star \mapsto \pm \end{bmatrix} \rangle$$

$\varnothing$

$\varnothing$

Table 3.6: The Latin Determiner `iste` 'this'

		MASC	FEM	NEUT
SG	NOM	iste	ista	istud
	GEN	istius	istius	istius
	DAT	isti	isti	isti
	ACC	istum	istam	istud
	ABL	isto	ista	isto
PL	NOM	isti	istae	ista
	GEN	istorum	istarum	istorum
	DAT	istis	istis	istis
	ACC	istos	istas	ista
	ABL	istis	istis	istis

Table 3.7: Mordvin Nouns: `val'ma` 'window'

	SG		PL	
	DEF	INDEF	DEF	INDEF
NOM	val'maś	val'ma	val'mat'ńe	val'mat
GEN/ACC	val'mańt'	val'mań	val'mat'ńeń	val'mań
DAT/ALL	val'mańt'eń	val'mańeń	val'mat'ńeńeń	val'mańeń
ELA	val'mastońt'	val'masto	val'mat'ńeste	val'masto
TRSL	val'maksońt'	val'maks	val'mat'ńeks	val'maks

Table 3.8: Sanskrit dvau ‘two’

$\langle x : \Diamond\ominus : \begin{bmatrix} \text{GENDER} & : & \star \\ \text{NUM} & : & 2 \\ \text{CASE} & : & \star \end{bmatrix} \rangle$
$\varnothing$
$\sharp x \doteq 2$

Numerals Numerals behave basically either as adjectives or as nouns. For example, Russian numerals (larger than four) assign genitive case to their complement (see Section 4.5). When they are nouns, they typically only inflect for case. (As nouns, their cardinality is fixed.) If they are adjectives, they are typically restricted in their ordering with respect to other adjectives. They must be structurally higher. This can be explained by the observations above that number closes the structure and turns it into an NP. So, we expect that the typical word order is

Determiner + Numeral + Adjectives + Noun

In many inflectional languages, there is also number marking. This works in essentially the same way as numerals, with two important differences: (a) it may be iterated at every word, while the numeral is not repeated again, (b) number marking is comparatively rudimentary, distinguishing at most the cardinalities *1*, *2*, *3*, *a few* and *many* (where ‘many’ means: more than otherwise morphologically expressed) (see [41]). If languages have number marking, the numerals cannot choose to which number they belong: they are inflected according to the number they express. So, in Sanskrit, the number ‘one’ is in the singular, the number ‘two’ in dual and the others in the plural. With increasing number the distinctions in case decrease in number. (See [16] for these facts.) Similarly in Latin, where the numbers from four onwards do not inflect any more.

Pronouns Finally, we turn to pronouns. Pronouns are full nouns and therefore inflect like nouns. Moreover, there exist definite and indefinite pronouns. The pronouns are by default definite; the indefinite pronouns are usually specially marked and derived from some definite pronouns. Typically the paradigms for pronouns are richer than those for nouns. First, other than nouns, which are invariably third person (at least syntactically), pronouns can be 1st, 2nd or 3rd person. Thus there is an additional dimension. Furthermore, pronouns can have richer paradigms, or even different paradigms. In English, for example, the pronouns also show a distinction between nominative and accusative, while ordinary nouns do not.

In many languages only 3rd person pronouns distinguish between genders. But this is not necessarily so. In Ngala, a Papuan language (see [34], page 80) the gender distinction is also present in the 1st and 2nd person. Furthermore, in certain ergative languages pronouns may actually be inflected along a nominative-accusative scale, while the other nouns are inflected along the absolutive-ergative scale (see [28]). The pronouns show syncretism in the S and A, while nouns exhibit syncretism with respect to S and U. In fact, Foley reports ([34], page 105) that the Papuan languages Yimas and Anggor distinguish all three functions, which gives support to this idea.

Another variety of pronouns are the interrogative pronouns and the relative pronouns. Interrogative pronouns behave in much the same way as the words of their category. But there is a difference that puts them into the same category as pronouns. Nominal interrogatives inflect not only for case and number, like nouns, but also in gender, since there must be pronouns for all genders. However, as the interrogative by its very nature leaves open the gender (and number) of the things asked for, nominal interrogatives tend to inflect less than an ordinary noun. Adjectival interrogatives on the other hand show the full distinction by agreement rules like adjectives. Adjectival interrogative pronouns are therefore likely to be used as relative pronouns, though the semantics of the two are distinct. They are distinct for example in German. The German interrogative `wer` ('who') does not inflect for case or number, while the relative pronoun `der` ('who') is in fact the same as the definite determiner. In Latin, the nominal interrogatives `quis` show a distinction only between neuter and non-neuter (which is either feminine or masculine) and exists only in the singular, while the adjectival interrogative `qui` has all three genders and all numbers. The same considerations hold for the indefinite pronouns (like `someone`, `something`). The relative pronoun is actually an entity that does not fit the schema so established so far. For notice that it engages in a case relation with the lower verb and returns a modifier of the noun (or noun phrase). Therefore, as is easily verified, it shall assume the role of an argument *twice* for the same referent, which yields and illegal argument structure.

Notes on this section. The symbol ⋆ has a rather complex role in the present framework. If it were to mean 'undefined' then we should say that Hungarian adjectives are undefined for gender. However, in accordance with the theory of layers of Chapter 5 we will speak of agreement and case as being assigned in layers. At the innermost layer we find the generic value ⋆, which may or not be converted at some outer layer into a different name. The difference between the concept of generic versus undefined can perhaps be explained with the concept of null case versus no case. It has been argued for example by Fanselow [32] that infinitives rather assigning no case to their subjects do assign a so-called null case. If we identify null case with the case called ⋆, this can still explain why German infinitives do not have overt subjects in infinitives: overt nouns must have case different from ⋆. The nearest equivalent to undefined case is the demoted subject of a passive. Here the verb does not even assign ⋆ to its former subject; rather, the case becomes undefined.

3.4 The Verbal Group

The verbal group is composed out of several layers, like the nominal group. There are strictly verbal dimensions, and there are dimensions that the verb shares with the nominal arguments. The verbal dimensions are usually inside the nominal dimensions. We will therefore discuss the latter in the next section, concentrating here on the verbal dimensions. These include: *voice*, *tense*, *aspect*, *mood*, *polarity* and *force*. There are many verbal categories that come to mind here, for example frequentatives and intensifiers, causatives and other elements, which are usually treated as derivational affixes. This means that they are added even before the voice. Of course, our present framework is capable of treating them as well, but we have opted here to leave them out of discussion. We will also have very little to say about aspect and mood, and so they will be only discussed in passing. Like with nouns, we assume that verbs are stored in the lexicon with the least number of features preassigned. The role of the morphology (or syntax) is to introduce a value where none is present yet. The features in turn see to it that certain morphemes cannot be added more than once and that they are added in the right order.

We can roughly distinguish four parts in the verbal inflectional system. The innermost part is concerned with the administration of the arguments. It determines the transitivity and the the voice of the verb. The second part is called the Tense-Aspect-Modality complex in [38]. After that comes a layer that is very much *complementizing* in nature. It consists of the *polarity* and the *force*. The outermost layer is the agreement layer. It determines the case assignment properties as well as the agreement properties. This organization is quite universal across languages. The voice system is innermost. Although conceptually this is not necessary, languages put the voice system at the innermost layer of the verb. This has to do with the fact that voice produces verbs of different transitivity. This influences the agreement pattern of the verb. Hence, agreement is outside of voice. This much seems clear without further argumentation. However, the relative position of voice with respect to tense, aspect and modality is less clear. There usually are subtle interactions between voice with aspect. We find, for example, that the passive in modern Indo-European languages (and not only there) is formed by using the perfect active stem. These facts need close analysis, but will be put aside here (but see below for a brief discussion). We will survey the structure of the verb from inside out. Thus, we discuss first voice.

Voice systems have different function. The most salient one is the promotion of oblique arguments. This can be quite important. In Tagalog, for example, only subjects can be relativized. This looks like a severe restriction. However, at the same time Tagalog has an elaborate system to promote any verbal argument to subject position. Similarly KinyRwanda, where only subjects and objects can be relativized (see [39]). There is an apparent exception to this rule, namely impersonal constructions. In Ute (an Uto-Aztecan language, see below for the data), there is a kind of passive that simply removes the subject without any promotion of the object. Similarly the Finnish passive ([52]). What is interesting about Ute is that the subject triggers number agreement on the verb, and that this agreement morpheme is inside the passive morpheme. We will see that there is a natural explanation within the system advocated for here. Even with subject agreement, the passivized verb together with the subject agreement is an intransitive verb. If passive consists in reordering the argument structure, then the passive of Ute is no different from other passives. We note here, however, that it might be useful to call this construction not *passive* but *impersonal*.

The next layer after the passive is the TAM complex. It too is structured. First comes the aspect, then the modality and last is the tense. Evidence for this ordering comes for example from the ordering of Creole auxiliaries. According to Bickerton (quoted from [39]) the Hawaii Creole orders the auxiliaries as follows.

(Anterior) (Modal) (Durative) Verb

Similarly, in English we find the following sequence (after [39])

(tense) (modal/irrealis) (have/perfect) (be/durative) Verb

In German (as in many modern Indo-European languages), the perfect is formed through the use of auxiliaries which are added to the perfect participle. Semantically, the perfect participle denotes a *state* (and can therefore be used as an adjective). It is *aspectual*. In Latin, however, there is a distinction between the active perfect, which is formed through a single morpheme, and the passive perfect, which is formed analytically. The perfect in the active takes an event variable and returns an event variable, while the passive returns a *state*. This accounts for the fact that in the active it is formed through a single morpheme. After tense comes mood. The predicate is formed after the tense marker has been added. Moreover, we assume that infinitives are formed at this moment as well. So, the tense category comprises the standard tenses (*present*, *past*, *future*) and also *indefinite*. Infinitives are generally tenseless but do have aspect (many languages have infinitives according to the various aspects). So, one must carefully distinguish between tense being undefined (bare verbs) and tense being indefinite (infinitives).

Finally, after tense, the agreement markers are added. The agreement markers not only add the agreement suffixes but also the case requirement for the major actants. This accounts for a number of facts. First, the direction in which the agreement suffix is found is often different from the direction in which the arguments are identified. For example, subject agreement in English is to the right of the verb, but the subject must appear on the left. In many languages, where the word order is principally free, the agreement markers appear in a fixed order. This is explained by assuming that the verb first of all selects basic actants without giving them a case requirement. The transformation of the case and agreement system offers the possibility to reassign the directionality of the assignment. The reason why case assignment is not fixed at the lowest level of the verb has another reason. We find, for example, that case assignment depends on various other factors, such as aspect, not to mention the obvious one, namely voice. Though this can also be achieved through transforming the case requirements, we will ultimately free the verb of any assignment properties of the verb with respect to its major actants (subject, actor and undergoer).

To summarize, we find the following sequence

Agreement Force Polarity Tense Modality Aspect Voice Root

The complex Voice-Verbal Root is simply a verb by nature. The complex

T+M+A+Vc+Root

we call a *predicate*. Finally, the F+P+T+M+A+Vc+Root is simply a sentence. Languages differ with respect to various parameters: whether the various groups are realized by morphemes or by distinct words, where the word boundaries are, and with respect to the directionality of the realization. For example, In Latin, the whole complex is a single word, only that polarity and force other than the default ones must be expressed separately. In Creole languages, all of the elements are expressed by distinct words.

The Verbal Root Verbal roots specify the term arguments of the verbal root as well as prepositional objects. The most important feature of roots is that the term arguments are exported, not imported. This is necessary for several reasons. One is the fact that otherwise the formation of complex predicates does not work properly. Recall from Section 2.5 that infinitives export the subject. This allows to use polyadic merge.

There is no case assignment to term arguments; the exported arguments only have grammatical relations. For example `kill` has the following argument structure.

(3.29)

`/kill/`

$\langle e : \triangle\ \bigcirc : \begin{bmatrix} \textsc{cat} & : ev \\ \textsc{voice} : & \star \end{bmatrix} \rangle,$ $\langle x : \triangle \bigcirc : [\textsc{gr} : \mathit{1}] \rangle,$ $\langle y : \triangle \bigcirc : [\textsc{gr} : \mathit{2}] \rangle.$
e
$\mathsf{kill}'(e); \mathsf{act}'(e) \doteq x;$ $\mathsf{thm}'(e) \doteq y.$

Voice Let us start with *voice*. We have earlier spoken of the fact that untensed verbs do not assign case to their subject. We will advance the thesis here that they do not even assign case to their direct objects. Let us take an example. The passive in English simply turns a transitive verb into an intransitive one. The former subject becomes an optional adjunct. We assume that since it can only in the form of a PP opened by `by` that the passive actually assigns a special case called *by*. The fact that we make no distinction between case marked DPs and PPs will be subject to an extended discussion in a later chapter. Passive operates on several variables at once, hence we need polyadic merge for it. We propose the following semantics for passive (of transitives):

(3.30)
$$
\begin{array}{|l|}
\multicolumn{1}{c}{/\textsc{pass}/} \\
\hline
\langle e : \blacklozenge\olessthan : \begin{bmatrix} \textsc{cat} & : ev \\ \textsc{voice} & : \star \mapsto pass \end{bmatrix} \rangle, \\
\langle x : \triangledown\ogreaterthan : [\textsc{gr} : \mathit{1} \mapsto \star] \rangle, \\
\langle y : \lozenge\olessthan : [\textsc{gr} : 2 \mapsto \mathit{1}] \rangle, \\
\langle z : \triangledown\olessthan : [\textsc{case} : \star \mapsto by] \rangle, \\
\\
\hline
\varnothing \\
\hline
x \doteq z. \\
\hline
\end{array}
$$

We can identify passive in English with the ending `-ed`. Notice that the passive is a transformer. It merges with the verb by identifying the event variable, the actor and the undergoer variable. The underlying semantics is empty. We assume that in the sentence `He is being killed.`, `He` is subject but nevertheless the theme and not the actor. Then `killed` will have the argument structure

(3.31)
$$
\begin{array}{|l|}
\multicolumn{1}{c}{/\mathtt{killed}/} \\
\hline
\langle e : \vartriangle \bigcirc : \begin{bmatrix} \textsc{cat} & : ev \\ \textsc{voice} & : pass \end{bmatrix} \rangle, \\
\langle x : \triangledown\ogreaterthan : [\textsc{case} : by] \rangle, \\
\langle y : \vartriangle\bigcirc : [\textsc{gr} : \mathit{1}] \rangle. \\
\hline
e \\
\hline
\mathsf{kill}'(e); \mathsf{act}'(e) \doteq x; \\
\mathsf{thm}'(e) \doteq y. \\
\hline
\end{array}
$$

Notice that the passive is a transformer. Notice also that the term argument for subject is no longer exported but imported. We shall see later that once the arguments are identified under case the vertical direction is reversed: they are then imported. This also means that for the first time the directionality of these arguments is specified. For if a referent is only exported no directionality is specified;

only when it is imported does directionality come in. We shall see that there is evidence for this. English verbs select their complements to the right.

To the root, either PASS is suffixed (for transitive verbs) or ACT:

(3.32)

/ACT/
$\langle e : \blacklozenge\olessthan : [\textsc{voice} : \bigstar \mapsto \mathit{act}]\rangle$
$\varnothing$
$\varnothing$

This analysis turns the subject into a PP opened by by. Notice that the notion of 'adjunct' and the like play no role, neither is the fact that the argument is optional represented. This can easily be done. Notice also that the grammatical relation of the subject is lost. It has been traded for the case. In terms of Relational Grammar, the subject has therefore become a *chômeur*. This means that it can no longer partake in any valency changing operation. It remains to see how the by-phrase is manufactured. there are is a function act′ selecting the actor of an event. We now assume that by is (among other) a preposition producing adverbs that can modify passivized verb phrases. Hence, the semantic structure for this word is

(3.33)

/by/
$\langle x : \lozenge\ogreaterthan : [\textsc{case} : \mathit{acc} \mapsto \mathit{by}]\rangle.$
$\varnothing$
$\varnothing \doteq x$

This shows only one example of the possibilities. Indo-European languages mainly use passive to promote objects to subject. There is passive in German (using the verb kriegen) which promotes beneficiaries to subject. Moreover, Ancient Greek among other languages has a *medium*. We will not discuss this construction here, however. In Section 4.5 we shall also exhibit some phenomena that take place before voice is added. This provides interesting evidence against the principle of lexical integrity.

Tagalog also uses passive to promote to subject. This is done by means of a prefix. However, there are several other prefixes, depending on which type of argument gets promoted. The former subject becomes a chômeur and assumes genitive case. (See [60] for the data below.) [4]

(3.34) `B-um-ili ang-lalake ng-isda sa-tindahan.`
PERF.ACT-buy NOM-man GEN-fish DAT-store

(3.35) `B-in-ili-∅ ng-lalakeang-isda sa-tindahan.`
PERF-buy-OV GEN-man NOM-fish DAT-store

(3.36) `B-in-ilh-an ng-lalake ng-isda ang-tindahan.`
PERF-buy-DV GEN-man GEN-fish NOM-store
The man bought fish at the store.

(3.37) `Ip-in-am-bili ng-lalake ng-isda ang-pera.`
IV-PERF-buy GEN-man GEN-fish NOM-money
The man bought fish with money.

(3.38) `I-b-in-ili ng-lalake ng-isda ang-bata.`
BV-ACT-buy GEN-man GEN-fish NOM-child
The man bought fish for the child.

We see that there are five voices: active, direct object promotion, indirect object promotion, instrument promotion and beneficiary promotion. We remark here that Tagalog allows free ordering of the arguments, so that any permutation of the arguments in the above sentences is grammatical. However, the verb must keep its first position.

KinyaRwanda is a language where arguments are promoted to direct object. According to [39], Ute (Uto-Aztecan) exemplifies a nonpromotional passive, namely simply deletion of the actor without any promotion of the object. However, it is noted that the actor still imposes its number agreement on the object.

(3.39) `ta’wá-ci sivą́ą̨tu-ci pax̂á-pų̨ga.`
man-SUB goat-OBJ kill-REM
The man killed the goat.

(3.40) `siv’ą̨ą̨tu-ci pax̂á-ta-pų̨ga.`
goat-OBJ *kill* – PASS – REM
Someone killed the goat.

(3.41) `táata’wá-ci sivą́ą̨tu-ci pax̂á-qa-x̂a.`
man-SUB-PL goat-OBJ kill-PL-ANT
The men killed the goat.

[4] There is a slight problem with the segmentation into morphemes, which I was unable to resolve.

(3.42)
```
siv'ąątu-ci  paxâ-q-ta-pųga.
```
goat-OBJ kill-PL-PASS-REM
Some people killed the goat.

This can be easily captured by analyzing Ute passive simply as a bound subject argument.

(3.43)

/ta/
$\langle e : \Diamond \circleddash : [\text{VOICE} : \star \mapsto \textit{pass}]\rangle$, $\langle x : \triangledown \circleddash : \top\rangle$.
$\varnothing$
$\varnothing$

Notice by the way that the plural marking is inside the passive marking. This is to be expected, since after the passive is affixed, the subject is expelled from the argument structure and can no longer trigger agreement. For that reason, we call the morpheme *argument saturator*. Generally, all relation changing operations as studied in Relational Grammar take place at the voice level. An example is possessor raising. In German, the possessor of a direct object may be promoted to a dative argument under the condition that the latter is also the beneficiary of the event.

(3.44) `Der Kater hat mein Frühstück weggegessen.`
the tomcat-NOM has my-ACC breakfast-ACC away-eaten
The tomcat has eaten my breakfast.

(3.45) `Der Kater hat mir das Frühstück weggegessen.`
the tomcat-NOM has me-DAT the-ACC breakfast-ACC away-eaten
The tomcat has eaten my breakfast.

(3.46) `Der Kater hat mir mein Frühstück weggegessen.`
the tomcat-NOM has me-DAT my-ACC breakfast-ACC away-eaten

Possessor raising is done by means of an empty operator, POSSR.

(3.47)

/POSSR/
$\langle e : \Diamond \circleddash : [\text{TRS} : +]\rangle$, $\langle x : \Diamond \oplus : [\text{GR} : 2]\rangle$, $\langle y : \triangle \bigcirc : [\text{GR} : 3]\rangle$.
$\varnothing$
$\text{ben}'(e) \doteq y : \text{belong}'(x, y)$.

Figure 3.6: The English Past Operator

$$
\begin{array}{|l|}
\multicolumn{1}{c}{/\textsc{past}/} \\
\hline
\langle e : \Diamond\oslash : \left[\begin{array}{lll} \textsc{voice} & : & \upsilon \\ \textsc{asp} & : & \alpha \\ \textsc{tense} & : & \star \mapsto \mathit{past} \end{array}\right]\rangle \\
\hline
\varnothing \\
\hline
\mathsf{time}'(e) < \mathsf{now}. \\
\hline
\end{array}
$$

This accounts well for the fact that possessor raising does not preclude the overt marking of the possessor.

Mood There is only a small number of moods: the indicative, the conditional, the irrealis, the potential and the optative. We will not be exhaustive in the classification here. Notice that mood (as a morphological category) must be distinguished from modal verbs, whose number is usually far greater. Their place in the verb cluster can be different (namely lower). Though the semantic analysis of mood is rather difficult, the argument structure is rather easily spelt out.

$$
\begin{array}{|l|}
\multicolumn{1}{c}{/\textsc{pot}/} \\
\hline
\langle e : \Diamond\oslash : [\textsc{mood} : \star \mapsto \mathit{pot}]\rangle \\
\hline
\varnothing \\
\hline
\mathsf{pot}'(e) \\
\hline
\end{array}
\qquad (3.48)
$$

Here $\mathsf{pot}'(e)$ says nothing more than that e is a potential event.

Tense We distinguish *definite* tense from *indefinite* tense; the definite tenses are further distinguished in *present*, *past* and *future*. A verb in indefinite tense is called an *infinitive*. Semantically, the tense operator adds in the definite case the temporal anchoring of the event. Our analysis will be rather shallow. A more detailed analysis will follow in Chapter 4. We assume that we have an indexical now, giving us the current time, and a function time from events to intervals. $I < \mathsf{now}$ means that I ends before now. Likewise, future, present and indefinite tense are treated. In the future, we replace $\mathsf{time}'(e) < \mathsf{now}$ by $\mathsf{time}'(e) > \mathsf{now}$, and for the present we write $\mathsf{time}'(e) \circ \mathsf{now}$, stating that $\mathsf{time}(e)$ overlaps with now. Finally, in the indefinite tense the body is empty.

Polarity and Force After tense has been added, the predicate is formed. To form a sentence, two more things are needed: the polarity and the force. There are two kinds of polarities: positive and negative. Basically, polarity determines whether the proposition is accepted or rejected. There are mainly three kinds of forces: stative, interrogative and imperative. Only after polarity and force are added we find the agreement markers. This follows from various observations, one being that case marking also depends on the polarity and the force. In Finnish, accusative complements have a special form when they are complements of an imperative verb. Moreover, many languages (including Finnish) distinguish accusative and partitive, and the choice between accusative and partitive is determined among other factors by the polarity of the verb.

Finnish is provides good evidence for the structure of the polarity/force complex. First of all, Finnish has a negation verb, which is inflected for person and number of the subject. Furthermore, this negation verb carries the force marker. Hence, we can note that morphologically, Finnish breaks the verbal complex if necessary after the TAM complex. Force-Polarity and agreement are fused into one word. In Indo-European languages we find that agreement is on the verb while negation is an uninflected word. However, English is also interesting because negative polarity and force distinct from stative must be expressed using the particle do. This can be explained as follows. Force and polarity must precede the TAM complex in English. Thus, the sequence is as stated above. The morphology of English is as follows. F+P may be unexpressed if default (positive and stative) or if the force is imperative. [5] If different, they must be expressed using some word. English has do. We then get the following sequence:

`did`	`not`	`read`
`shall`	`not`	`have read`
(F+T)	P	(A+)V

Our model does not capture the full range of facts here as easily. However, see Section 3.7 for an explanation.

[5] This is quite universal.

Turkic languages also have negation suffixes on the verb itself. The negative suffix mA (where A represents a or e, depending on vowel harmony). This suffix is added before tense and agreement and following passive, reflexive, reciprocal and causative suffixes if present (see [58]). Here is an example.

	Hasan	kitab-ı	oku-ma-dı
(3.49)	Hasan	book-ACC	read-NEG-PAST
	Hasan did not read the book.		

The negation on the verb does nothing but to switch the polarity of the sentence. The actual negative meaning must be inserted later. This is an inevitable consequence of the fact that the negative marker takes scope over elements that are outside of the verb. In Turkish there is also a negation verb değil which functions just like the Finnish negation verb or as a constituent negator. We also find the construction where negation is outside of the tense:

	Ben	iş-im-i	bırak-acak	değil-im
(3.50)	I	work-POSS:1.SG-ACC	leave-FUT	NEG-1.SG
	I shall not leave my work.			

So, the placement of negation can vary. It is often hard to decide whether or not tense is taken to be inside of negation. This may explain the ambiguity here. These two negation markers can cooccur. When the two cooccur, the meaning is not positive but negative. We shall therefore assume that in addition to setting the polarity from $\star$ to $-$ they can also show ‘polarity agreement’ in simply mapping $-$ to $-$.

3.5 Verbal Agreement

In this section we will discuss the last layer of the verbal group, namely *agreement*. Generally, verbs have up to three arguments, with very few exceptions. All other actants are adverbials, which means that they can be added freely and in any number. A specific conclusion is that verbs agree with up to three arguments. However, these arguments need not be the arguments of the bare verb. We only mention here causatives, infinitival complements and possessor raising as typical sources of additional arguments with which the verb can show agreement. Now, agreement is added in three shells. We assume that the order is as follows.

AgrS AgrIO AgrDO V

Here, AgrS denotes the subject agreement, AgrIO the indirect object agreement and AgrDO the direct object agreement. Deviances are possible. We will confine ourselves first to the discussion of subject and object agreement. AgrDO can be added to any verb which has an argument with an [GR : 1] entry. When AgrDO is attached, it adds [CASE : *acc*] and (optionally) changes the directionality of the assignment. [6] Typically, AgrDO has a number of forms depending on the NP. Hence, we find agreement in number, class, definiteness, and person. Next the AgrIO is attached. It requires the verb not to case mark the subject (so that AgrS has not been added yet) and to be either intransitive or transitive and have an argument with accusative case (therefore AgrDO has been added already); the nature of AgrIO depends on the type of third argument, whether it is beneficiary, or carries dative case etc. Once it has found an argument to which it can attach, it acts similarly to AgrDO. [7] At last, AgrS is added. It has the same requirements as AgrIO.

All the parametric variation is in the argument structure of AgrS, AgrIO and AgrDO. For example, in many languages subject agreement is null if the verb has indefinite tense, others show overt agreement though of a different nature than in the finite tense, as in Hungarian. We say therefore that AgrS requires definite tense, and that there is a special zero agreement suffix for indefinite tense. Furthermore, many languages exhibit case on an ergative-absolutive scale. This can be captured by assuming that AgrDO assigns absolutive case and that AgrS assigns absolutive case to intransitive subjects (that is, when the GF is S) and egative to transitive subjects (that is, when the GF is A). This captures a system where the agreement is on a nominative-accusative scale. However, by changing the argument structure of AgrS and AgrDO somewhat we can also capture agreement on an ergative-absolutive scale. Finally, some languages make a difference as to whether the complement is a pronoun or not. That can also make a difference in the case marking. In general, there is the possibility to treat the case marking of Acc/Nom versus Erg/Abs as a property of nouns. Another possibility is to assume that AgrS and AgrDO simply depend on the case assignment properties whether

[6] Obviously, if the language has no case marking (like Zulu, or even English) then the agreement markers will not introduce any case feature.

[7] The present system is such that AgrIO turns out to be nonobligatory. This is an accident and not an inherent feature of the model. Since we will focus on subject and object agreement, we leave that issue aside here.

the argument is a pronoun. We favour the latter strategy for various reasons. (1) The case distinctions in pronouns is richer than in full nouns. English is an example. Now, rather than assuming a nominative-accusative case distinction for all nouns, we will say that nouns have no case (or a single case only), while pronouns have two cases. (2) Certain languages allow pronouns to incorporate into a preposition. Therefore, the preposition has different forms depending on the person and number of the complement. (It is sometimes said that the prepositions inflect for person, but this is a rather infelicitous use of the terms.) If we allow prepositions to distinguish between a pronominal complement and a full complement, then these facts can be accounted for rather easily.

Therefore, rather than using AGRS and AGRDO, we will use AGRS, which is agreement subject (of intransitives), AGRA and AGRU. Accusative verb marking consists in grouping AgrS and AGRA under AGRS; then AGRDO is the same as AGRU. Ergative verb marking consists in grouping AGRS and AGRU to what might be called AGRABS and AGRA under AGRE. We summarize these different morphemes in the Figures 3.7 and 3.8. (Recall that $\top$ simply means that the entry is other than $\star$.) Some notes are in order to explain the shape of these argument structures. We must assure that AGRU and AGRA attach to transitive verbs and AGRS to intransitive verbs. In accusative languages, this is unproblematic, since we may treat a transitive verb with AGRA simply as if it it were intransitive. In ergative languages, this is not the case. Here, the intransitive verb assigns absolutive to its subject, while the transitive verb assigns ergative. Hence, we have introduced a feature E with values $+$ and $-$ which monitors whether what we have is an intransitive verb or a transitive verb with AGRU added. By default, E is set to $-$. Now, in accusative languages we must make sure that AGRU is added before AGRA. Hence, AGRU can attach only if there is a subject to which the verb still does not assign a case.

In many languages the case assignment not only depends on the grammatical function but also on other factors. Quite common is the distinction between accusative and partitive for direct objects, which depends among other on the aspect. Georgian shows a rather systematic change in case assignment depending on aspect. (See [56] for a discussion of the implication of this fact for the checking theory outlined in the Minimalist Program.)

Agreement markers can be clitic pronouns, affixes, or simply inflections. Often, in an inflectional system, the object agreement and the subject agreement cannot be separated. However, all this is no limitation of the present system.

Figure 3.7: Types of Agreement Suffixes (Accusative Languages)

/AGRS/

$$\langle e : \Diamond\olessthan : \begin{bmatrix} \textsc{force} & : & \top \end{bmatrix}\rangle$$

$$\langle x : \Diamond\olessthan : \begin{bmatrix} \textsc{gr} & : & \mathit{1} \\ \textsc{case} & : & \star \mapsto \mathit{nom} \end{bmatrix}\rangle$$

$\varnothing$

$\varnothing$

/AGRA/

$$\langle e : \Diamond\olessthan : \begin{bmatrix} \textsc{force} & : & \top \end{bmatrix}\rangle$$

$$\langle x : \Diamond\olessthan : \begin{bmatrix} \textsc{gr} & : & \mathit{1} \\ \textsc{case} & : & \star \mapsto \mathit{nom} \end{bmatrix}\rangle$$

$\varnothing$

$\varnothing$

/AGRU/

$$\langle e : \Diamond\olessthan : \begin{bmatrix} \textsc{force} & : & \top \end{bmatrix}\rangle$$

$$\langle x : \Diamond\olessthan : \begin{bmatrix} \textsc{gr} & : & \mathit{2} \\ \textsc{case} & : & \star \mapsto \mathit{acc} \end{bmatrix}\rangle$$

$$\langle x : \Diamond\olessthan : \begin{bmatrix} \textsc{gr} & : & \mathit{1} \\ \textsc{case} & : & \star \end{bmatrix}\rangle$$

$\varnothing$

$\varnothing$

Figure 3.8: Types of Agreement Suffixes (Ergative Languages)

/AGRS/

$$\langle e : \Diamond\otimes : \begin{bmatrix} \text{E} & : & - \\ \text{FORCE} & : & \top \end{bmatrix}\rangle$$

$$\langle x : \Diamond\otimes : \begin{bmatrix} \text{GR} & : & \mathit{1} \\ \text{CASE} & : & \star \mapsto abs \end{bmatrix}\rangle$$

∅
∅

/AGRA/

$$\langle e : \Diamond\otimes : \begin{bmatrix} \text{E} & : & + \mapsto - \\ \text{FORCE} & : & \top \end{bmatrix}\rangle$$

$$\langle x : \Diamond\otimes : \begin{bmatrix} \text{GR} & : & \mathit{1} \\ \text{CASE} & : & \star \mapsto erg \end{bmatrix}\rangle$$

∅
∅

/AGRU/

$$\langle e : \Diamond\otimes : \begin{bmatrix} \text{E} & : & \star \mapsto - \\ \text{FORCE} & : & \top \end{bmatrix}\rangle$$

$$\langle x : \Diamond\otimes : \begin{bmatrix} \text{GR} & : & \mathit{2} \\ \text{CASE} & : & \star \mapsto abs \end{bmatrix}\rangle$$

∅
∅

Figure 3.9: A Transitive Verb with Nominal (left) and Verbal (right) Complement

/seh–$^{\checkmark}$/

$\langle e : \triangle\ \bigcirc : \star\rangle$, $\langle x : \triangledown\otimes : [\text{GR} : \mathit{1}]\rangle$, $\langle y : \triangledown\otimes : [\text{GR} : \mathit{2}]\rangle$.
e
see′(e) act′(e) $\doteq x$ thm′(e) $\doteq y$

/helf–$^{\checkmark}$/

$\langle e : \triangle\ \bigcirc : \star\rangle$, $\langle x : \triangledown\otimes : [\text{GR} : \mathit{1}]\rangle$, $\langle y : \triangledown\otimes : [\text{GR} : \mathit{3}]\rangle$, $\langle e' : \triangledown\otimes : [\text{TENSE} : inf]\rangle$.
e
help′(e); act′(e) $\doteq x$; act′(e') $\doteq y$; thm′(e) $\doteq e'$; ben′(e) $\doteq y$.

Figure 3.10: Subject Agreement (Elaborate Version)

$$
\begin{array}{|l|}
\multicolumn{1}{c}{/\textsc{agrs}/} \\
\hline
\langle e : \Diamond\oslash : \left[\begin{array}{lcl} \text{VOICE} & : & \upsilon \\ \text{ASPECT} & : & \alpha \\ \text{MOOD} & : & \mu \\ \text{TENSE} & : & \tau \\ \text{FORCE} & : & \varphi \\ \text{POL} & : & \pm \end{array}\right]\rangle \\
\langle x : \Diamond\oslash : \left[\begin{array}{lcl} \text{PERS} & : & \pi \\ \text{NUM} & : & \nu \\ \text{CLASS} & : & \kappa \\ \text{DEF} & : & \delta \\ \text{GR} & : & 1 \\ \text{CASE} & : & \star \mapsto nom \end{array}\right]\rangle \\
\hline
\varnothing \\
\hline
\end{array}
$$

Figure 3.9 shows some lexical entries for German transitive verbs, one transitive verb selecting a nominal complement and one selecting an infinitival complement. Selection is uniformly to the right. In general, subject agreement functions as given in Figure 3.10. It selects uniformly to the left an event and a subject. However, while the directionality of the selection of *e* triggers the place of subject agreement, the directionality of the subject agreement will supersede the rightwardness of the subject selection of the verb. AgrS has distinct forms depending only on the number and person of the nominal argument. The form that AGRS assumes, that is to say its exponent, could in principle depend in its form on all the variables involved in the list. However, in German it only depends on the number and person. In Hebrew it also depends on the gender, in Latin and Ancient Greek on the voice. This seems to be the extreme.

It would be too much to survey all the agreement possibilities. First of all, many languages have only agreement with one argument, the subject (virtually all Indo-European languages, Arabic and Finnish). Many languages allow double agreement (Hungarian ([93]), Mordvin ([55]), Zulu ([102]) and some threefold

Table 3.9: Hungarian `látni` 'to see'

			$\delta_U \rightarrow$ INDEF	DEF
ν_A, π_A	SING	1	`látok`	`látom`
$\downarrow$		2	`látsz`	`látod`
		3	`lát`	`látja`
	PLUR	1	`látunk`	`látjuk`
		2	`láttak`	`látjátok`
		3	`látnak`	`látják`

agreement (Basque, Georgian ([3]), Yimas ([34], page 94). Second, while many languages only have agreement in person and number (Latin ([9]), French ([104]), German etc.) there exist languages that show agreement in gender (Russian ([22]), Hebrew ([38]), Hindi ([67])) and class (Yimas, see [34]). The language Yimas has several sets of morphemes for displaying agreement with up to three arguments, with a special distinction being drawn between subject, actor, undergoer and recipient.

Single agreement is a well known phenomenon and bears little surprises. It is therefore of some value to give an example of double agreement. An example is provided by Hungarian verbs. They are interesting because they show (contrary to the claim by LaPointe in [63]) agreement with the object in only one property: definiteness of the object. So we have

(3.51) `Látok egy madarat.`
I see a bird.

(3.52) `Látom azt a madarat.`
I see that bird.

Moreover, there is a special form for the 1st singular subject plus 2nd singular object. The verbal paradigm we show is that of the verb `látni` (to see) in the present. (See, for example, [93].) The form `látlak` needs to be added, which means 'I see you'. Hungarian is also interesting for the following phenomenon.

(3.53) `Akarok látni egy madarat.`
want-SUB:1.SG-DO:INDEF see-INF DET:INDEF bird-ACC

I want to see a bird.

(3.54) `Akarom látni azt a madarat.`
want-SUB:1.SG-DO:DEF see-INF DET:DEF bird-ACC
I want to see that bird.

(3.55) *`Akarok látni azt a madarat.`
want-SUB:1.SG-DO:INDEF see-INF DET:DEF bird-ACC
I want to see that bird.

(3.56) *`Akarom látni egy madarat.`
want-SUB:1.SG-DO:DEF see-INF DET:INDEF bird-ACC
I want to see a bird.

As these four sentences show, the verb `akar` ('to want'), agrees with the complement of its infinitival complement. Moreover, all six permutations of the two verbs and the NP are (more or less) grammatically acceptable. This is a rather hard test. It shows that the complex verb becomes transitive when combined with a lower transitive verb. If the lower verb is intransitive (eg `futni` ('to run') then the complex is intransitive. Notice also that the following is fully grammatical.

(3.57) `Akarlak látni.`
want-SUB:1.SG-DO:2.SG see-INF
I want to see you.

We will return to this phenomenon below in Section 6.5. We shall note here that there is a close connection between Hungarian object agreement and clitic climbing in Spanish, which has standardly been analysed as a movement of a clitic. (However, for a critique see [2].)

In a related language, Mordvin, double agreement is even more elaborate. It encompasses person and number of the subject *and* the object. Moreover, it distinguishes all three agreement suffixes: AgrS, AgrA and AgrO. In Table 3.10 we show the paradigm of a transitive verb, `sodams` ('to know'). In Table 3.11 we show the same for an intransitive verb (the same verb again). The verb `sodams` can be used transitively in the sense of knowing something and of knowing someone (see [55]). Notice that certain forms are missing in this paradigm. These correspond to the reflexive use of the verb, when part of the subject is also part of the object. It would be interesting to see whether or not it is appropriate to use the third-third forms when subject and object are (partly) identical. We have

Table 3.10: Mordvin Double Agreement: sodams 'to know'

ν_A, π_A ↓	$\nu_U = sg, \pi_U$ → 1.SG	2.SG	3.SG
1.SG	–	sodatan	sodasa
2.SG	sodamasak	–	sodasak
3.SG	sodasamam	sodatanzat	sodasi(zé)
1.PL	–	sodatadiź	sodasińek
2.PL	sodasamiź	–	sodasink
3.PL	sodasamiź	sodatadiź	sodasiź
ν_A, π_A ↓	$\nu_U = pl, \pi_U$ → 1.PL	2.PL	3.PL
1.SG	–	sodatadiź	sodasiń
2.SG	sodamasiź	–	sodasit'
3.SG	sodasamiź	sodatadiź	sodasińźe
1.PL	–	sodatasiź	sodasińek
2.PL	sodasamiź	–	sodasink
3.PL	sodasamiź	sodatadiź	sodasiź

Table 3.11: Mordvin Single Agreement

NUM	PER	
SG	1	sodan
	2	sodas
	3	sodi
PL	1	sodatano
	2	sodatado
	3	sodit'

Table 3.12: Potawatomi Agreement Suffixes

ν_A, π_A ↓	ν_U, π_U → 1.SG	2.SG	3.SG	3.SG OBV	1.PL INC	1.PL EXC	2.PL	3.PL
SUBJ								
1.SG	–	unan	uk	–	–	–	unuko	ukwa
2.SG	yun	–	ut	–	–	yak	–	utwa
3.SG	t	uk	–	at	unuk	yumut	unak	–
3.SG.OBV	–	–	ukot	–	–	–	–	ukwat
1.PL.INC	–	–	at	–	–	–	–	–
1.PL.EXC	–	unak	uko	–	–	–	unak	–
2.PL	yek	–	ek	–	–	yak	–	ukwa
3.PL	wat	uk'wa	–	awat	unuk	yumut	unak	–

not been able to evaluate this. The analogous forms are also missing in the Georgian and the Potawatomi verbs. We show in Table 3.12 the agreement markers for Potawatomi, taken from [44]. The data is interesting insofar as it provides evidence for our hypothesis concerning double agreement. Potawatomi distinguishes in the first person plural an inclusive and exclusive. Consequently, we should expect a form for the 1st exclusive subject and 2nd object but no form for the 1st inclusive subject and 2nd object. This is indeed what we observe. (The obviative forms are taken out of discussion here. See Section 6.6 for a discussion.) Finally, we give an example from a head–marking language, Zulu. Here, free pronouns are not so much used. Instead, there is a verbal agreement system for subject and object. It is summarized in Table 3.13. The agreement prefixes are roughly the same, whether they are subject agreement or object agreement. Notice that this hardly causes confusion, since there is no object agreement without subject agreement. Threefold agreement is either with subject-object-indirect object (Georgian, see [3]), or with subject-object-beneficiary. The latter is widespread in Papuan languages, see [34]. The agreement with the beneficiary derives historically from serializing the verb with the verb 'give' as the following example from Yimas shows. (Yimas has a class system. The roman numbers indicate the number of the class.)

(3.58) `urangk ki-n-yara-nga-r-umpun.`

coconut-VI.SG U:VI.SG-A:3.SG-get-give-PERF-BEN:3.PL

Table 3.13: Zulu Agreement Prefixes

	SUBJECT	OBJECT
1.SG	ngi-	-ngi-
1.PL	si-	-si-
2.SG	u-	-ku-
2.PL	ni-	-ni-
CLASS 1	u-	-m-
CLASS 1A	u-	-m-
CLASS 2	ba-	-ba-
CLASS 2A	ba-	-ba-
CLASS 3	u-	-wu-
CLASS 4	i-	-yi-
CLASS 5	li-	-li-
CLASS 6	a-	-wa-
CLASS 7	si-	-si-
CLASS 8	zi-	-zi-
CLASS 9	i-	-yi-
CLASS 10	zi-	-zi-
CLASS 11	lu-	-lu-
CLASS 14	bu-	-bu-
CLASS 15	ku-	-ku-

He got a coconut for them.

In this context it is interesting to note that languages with double (or threefold) agreement must provide also some paradigm for verbs that have only one (or two) arguments. There are two strategies. The first is to have a completely separate paradigm. This is realized in Mordvin. The indefinite (and intransitive) verbs inflect as given in Table 3.11. The second strategy is to use the full paradigm but inflect with a dummy object. Usually, this is the third singular. We have met this before discussing the passive of Ute. (See also the discussion of Georgian in [3]. Here the omission of an argument is also signaled by a so-called *screeve marker*.) Notice that the same problem arises also in languages with only subject verb agreement, namely with the restricted class of impersonal verbs (weather verbs). In Indo-European languages, for example, the third singular is used throughout. Here is an example from German, Latin and English.

(3.59) `Es regnet.`

(3.60) `Pluit.`

(3.61) `It rains.`

In many languages, however, the agreement system has kept its connection with the pronoun system, and the agreement morphemes are simply optional. For example, Zulu has subject agreement and object markers; the latter are left out if no object is present. Likewise in Papuan languages. However, as is reported in [34], in the language Kewa all verbs must be specified whether they are egocentric or not, hence whether the beneficiary is the speaker or someone else.

(3.62) `ta-lepaa`
hit-PAST.PL.EGO
you all hit it (for me/us)

(3.63) `taa-tepaa`
hit-PAST.PL.ALTRO
you all hit it (for someone)

A lot of attention has been directed in the last ten years to the so-called noun incorporation of verbs (especially due to the influential book by Baker [6]). Recently, in [7], Baker has proposed a different analysis. We will not go into the details of this approach. Rather, we think that the facts established so far nourish the thesis that noun incorporation as far as we can judge it is nothing but a complex agreement system. The structure of Mohawk verbs, for example, is the following. There are tense markers, aspect markers and various derivational suffixes; furthermore, verbs have agreement suffixes for agents and for objects. Finally, right before the verb appears what looks to be a bare noun. For example,

(3.64) `s-a-hʌ-[i]tsy-a-hseruny-á-hnaʔ`
ITER-FACT-A:MASC.SG-fish-∅-prepare-∅-PURP-PUNC
in order for him to prepare fish

Apart from the particle for aspect we find the agreement marker for subject (or actor). Furthermore, the word for fish, `itsy`, has been incorporated into the verb, `hseruny` ('to prepare'). In [6] and [7], incorporation is analyzed as syntactic movement, but there are several facts that militate against such a view. First, the incorporated element may well appear once again in the full NP. In the example below, the noun `nákt` (bed), has been incorporated into the verb, but appears once again in the NP.

	`Uwári`	`ʌ-ye-nakt-anúhweʔ-neʔ`	`ne Sak rao-nákt-aʔ.`
(3.65)	Mary	FUT-S:FEM.SG-bed-∅-like-PUNC	NE Sak MASC.SG.PL-bed-NSF
	Mary likes Sak's bed.		

Furthermore, a noun that can be incorporated seems to denote perceptually simple categories, and so if a noun is actually more specific, the more specific term must be used in the NP rather than being incorporated into the verb. This is why it the phenomenon is elswhere referred to as classifier incorporation. An example is (Page 310):

	`Shaʔtéku ni-kuti`	`rabahbót`	`wa-hʌ-[i]tsy-a-hnínu-ʔ`	`ki`
	eight PART-S:Z.PL	bullhead	FACT-S:MASC.SG-fish-∅-buy-PUNC	this
(3.66)	`rake-ʔníha`			
	my-father			
	My father bought eight bullheads.			

Rather than talking about 'incorporation' we wish to advance the thesis that what we have here is a case of a highly developed agreement system. For the direct object the verb does not use the usual agreement system via prefixes showing only gender and number, but rather a purely semantical agreement system. The object is classified by means of nouns and this class is identified by the noun that defines the class. (This is parallel to agreement systems that use pronouns instead.) So, there is a class for beds, for babies, for chiefs and for fish. Rather than saying 'I feed the baby' a Mohawk speaker can say 'I baby-feed'. However, we have argued that classes tend to be perceptually simple, so a Mohawk speaker would not say 'I bullhead-buy' but rather 'I fish-buy bullheads'. Space is too limited to discuss this proposal in depth. Let us say first of all that it is absolutely not unusual that the agreement systems are different for different arguments of the verb (see for example Hungarian above). So, the fact that we have 'class'-agreement for direct objects and gender and number agreement for the other arguments is not disturbing for our theory. However, there still is a set of data that need to be accounted for. Look at the following contrast. (See [7], Page 316.)

(3.67) `Shako-núhweʔ-s ne owiráʔa.`
s:masc.sg.o:3.pl-like-hab ne baby

(3.68) `Ra-wir-a-núhweʔ-s.`

(3.69) s:masc.sg-baby-∅-likes-hab
*`Shako-wir-a-núhweʔ-s`

(3.70) s:masc.sg.o:3.pl-baby-∅-like-hab
He likes babies.

The agreement with the the object is realized in a reduced form or with the classifier agreement, but not both. While we expect that the last sentence is ungrammatical (because there can be only one agreement form per argument), it is not clear why we have a choice for the agreement system (pronominal versus classifier agreement). The solution is surprisingly simple. Let us go back to the above examples. What we observe is that the incorporated noun appears right next to the verb, inside the actual agreement system. Hence, we propose that incorporated nouns can saturate the undergoer (=object) argument (as in Ute). This happens prior to anything else. The verb becomes syntactically intransitive, and therefore object agreement is blocked. Alternatively, if the noun is not incorporated, it does not block object agreement, in fact, it enforces it. This explains the morphological data satisfactorily, but leaves open a number of other questions. For by the mechanics of the merge, an intransitive verb does not tolerate an object argument. Therefore, the resumption of the object in the clause would be ungrammatical.

Notes on this section. As we have remarked above, there is a theory which assumes that we do not have two but three underlying cases, s, a and u, but that languages usually group either s and a (accusative languages) or s and u (ergative languages). There are a handful of languages that actually distinguish all three.

Such a language is Pitta Pitta and Thalanji. (See [71] for a discussion.) Assignment of cases is in many languages sensitive to tense and mood. This is captured by the fact that agreement markers have a slot for tense and mood and other categories as well.

3.6 Possessives and Other Inflecting Categories

Possessives In many languages, possession is expressed by some special syntactic means. Indo-European languages use the genitive case, but other cases are possible for example in Finnish, where the possession is expressed using the adessive case (see [13]). In other languages, there are special suffixes or particles that express possession. We turn our attention to the latter kind of construction.

With respect to possession there are two basic possibilities depending on which we assume to be the subject. Therefore, there are two verbs expressing possession: `own` (or `possess`) and `belong to`. Even though (3.71) and (3.72) are synomyous, nevertheless there is a subtle difference.

(3.71) `Jacks owns this house.`

(3.72) `This house belongs to Jack.`

The difference pertains to the choice of subject. A suject is preferrably definite, and therefore (3.73) is favoured over (3.74).

(3.73) `Jacks owns a house.`

(3.74) ?`A house belongs to Jack.`

One might think that possessive expressions of the form *X's Y* should be assimilated to the construction *X possesses Y*. One reason is that in both structures *X* is the subject; for it is the specifier in both structures (using the terminology of GB). This, however, is only one choice and not the most likely one. Rather, we will assume that *X's Y* is parallel in structure to the sentence *Y belongs to X*. The reason is that we get a syntactic analysis of the following kind:

(3.75) Y [`belongs to` X]

(3.76) [X`'s`] Y

This is in fact the structure that we see in the majority of languages. Here, Y is the head of the possessive construction. Therefore, in the usual case marking languages Y is the element that receives the case that is assigned to the whole expression, while X simply receives the genitive. [8] The sequence *belongs to X* is an (optional) modifier. [9]

In order to survey the various possibilities, we assume an abstract element, which we write `belong`, whose meaning is a binary relation, belong'. We write $\mathsf{belong}'(x, y)$ to state that x is the possessor and y the possessee; so, x possesses y. As it is a binary relation, it has two arguments and therefore agreement with both arguments should in principle be possible. However, one should take careful note of the fact that the element expressing possession can be a case ending (many Indo-European languages), a special morpheme (Finno-Ugric languages), or a function word (French, Hindi). The structure is as follows.

(3.77) DP [`belong` DP]

The possessed things is the first argument of the element `belong`. Moreover, it is the only argument to which `belong` assigns case, if at all. However, the element expressing the fact of possession can also be a morpheme, in which case it forms part of another word. The above structure suggests that the element `belong` is affixed to the possessor rather than the possessee. This is in fact the most canonical way, and we find it in Indo-European languages, in Finnish, and in Zulu. Yet the other possibility, to mark the possessed, also occurs, namely in Hungarian and Finnish. [10] In the case of Hungarian we may either argue that we have a bracketing paradox, or we argue that the semantics of the suffix marking the possessor is actually different. Let us look at an example:

(3.78) `a ház-am`
the house-POSS:1.SG
my house

(3.79) `a város központ-ja`
the city centre-POSS:3.SG
the center of the city

---

[8]In fact, the genitive simply *is* the sign of possession.

[9]Of course, the syntax of possessives is more involved, since the addition of a possessive marks the noun phrase as definite. This has the effect that it acts like a determiner in English. Moreover, it cannot be iterated.

[10]So, Finnish offers both possibilities, as is shown below.

Figure 3.11: Possession in Hindi

/ka/

$$\left\langle x : \Diamond\olessthan : \begin{bmatrix} \textsc{class} & : & \star \\ \textsc{num} & : & \star \\ \textsc{case} & : & \mathit{obl} \\ \textsc{gr} & : & \mathit{1} \end{bmatrix} \right\rangle$$

$$\left\langle y : \Diamond\ogreaterthan : \begin{bmatrix} \textsc{class} & : & \star \\ \textsc{num} & : & \star \\ \textsc{case} & : & \top \\ \textsc{gr} & : & \mathit{2} \end{bmatrix} \right\rangle$$

$\varnothing$

$\mathsf{belong}'(x, y)$

(The suffixes have different forms depending on the stem but this is a morphophonological difference only.) We propose therefore to paraphrase `-am` rather as *I own* and similarly `-ja` as *he/she/it owns*. (Or better still, `-am` is really like an inflecting verb, which can take an optional argument. See the data below.)

Therefore, the argument structure of `belong` is as in Figure 3.11. We have chosen to spell it out for Hindi, which is the first language which we will discuss. In Hindi, there are two cases, *direct* and *oblique*. Noun phrases in the direct case can be on their own, but if put into the oblique a noun phrase must be governed by a postposition, for example `ka`, indicating possession. What is interesting here is that `ka` is not invariable, but takes different forms, reflecting the case, number and gender of the possessor. To see this we contrast the forms of `ka` with those of an ordinary adjective. Take the adjective `acchā`, *good*. It has four forms.

	MASC	FEM
DCT SING	acchā	acchī
OBL SING	acche	acchī
DCT PL	acche	acchī
OBL PL	acche	acchī

Now consider the following sentences.

(3.80) us strī kā beṭā
that woman's son

(3.81) us strī ke beṭe
that woman's sons

(3.82) us strī ke beṭe kā makān
that woman's son's house

(3.83) us ādmī kī bahnoṃ kā makān
that man's sisters' house

In each of these sentences replacing the particular form of kā by another yields an ungrammatical sentence. We conclude that the phrases of the form NP + kā show the same agreement pattern as adjectives, and that pattern is realized on kā, while the case of the noun phrase itself is fixed, namely OBL. The same holds for the other postpositions. Given that these postpositions have an argument structure consisting of two referents, with respect to which they are argument and adjunct, then according to our principles laid out earlier exactly this possibility arises. It is important to note that the agreement system for postpositions is not identical to that of nouns, but rather to that of adjectives, since they are adjuncts, not heads for both their variables.

In English, possession is marked by a special case, the genitive. Therefore, there is an element that has the argument structure shown in Figure 3.11, but it is the genitive marker. A genitive marker, being a case ending, does not show agreement at all. However, there are noteworthy exceptions to this rule. Such exception is reported for Awngi, a Cushitic language, in Hetzron [47] (for similar data in Tsakur, a Daghestanian language, see Boguslavskaja [14]). This language has three forms of a genitive marker, depending on whether the possessor is masculine singular or feminine singular or plural. Here are the examples.

(3.84) murí-w aqí
village-GEN.MASC.SG man
the man of the village

(3.85) murí-t guna
village-GEN.FEM.SG woman
the woman of the village

(3.86) murí-kw aq(ká)/gunagúná
village-GEN.PL men/women
the men/women of the village

See also data in [22] on Chamalal, a Caucasian language. In Jiwarli, an Australian language this function of the genitive is taken over by the dative (Austin [5]) and indeed we find that the dative exhibits some phenomena that we would expect only of the genitive. Another set of cases which have an articulated argument structure are the cases of location. These are found in many languages.

There are languages which have a special syntactic construction for the possession. In some Bantu languages (Sesotho, [25]), there are possessive pronouns which are simply made of two pronouns, one indicating the class of the possessor and the other indicating the class of the possessee. Take the following example

(3.68)	mọ-thọ	é-mọ́-họlọ	ọ-rata	∅-ntjá	ẹ́-ntlɛ	éa-haệ
	1	1-1	1	9	9	9-1
	person	big	(s)he-like	dog	beautiful	of his/her

(The) old person likes his/her beautiful dog.

(3.69)	ba-thọ	bá-bá-họlọ	ba-rata	li-ntjá	tsẹ́-ntlɛ	tsá-bona
	2	2-2	2	10	10	10-2
	people	big	they-like	dogs	beautiful	of-them

(The) old people like their beautiful dogs.

So, the possessive are formed by prefixing the pronoun of the class of the possessor by a prefix indicating the class of the possessee. This latter strategy of adding a possessee prefix is used throughout Bantu languages. We illustrate the phenomenon here with data from Zulu. In Zulu, possessor nouns may agree with the possessed. As note [102], the possessive agreement suffixes are derived from the corresponding verbal agreement suffixes (shown in Table 3.13) by simply adding an -a. Here are some examples.

Table 3.14: Zulu Possessor Marking

Class	Possessor	Possessed
Class 1	u-	wa-
Class 1a	u-	wa-
Class 2	aba-	ba-
Class 2a	ba-	ba-
Class 3	u-	wa-
Class 4	i-	ya-
Class 5	li-	la-
Class 6	a-	a-
Class 7	si-	sa-
Class 8	zi-	za-
Class 9	i-	ya-
Class 10	zi-	za-
Class 11	lu-	lwa-
Class 14	bu-	ba-
Class 15	ku-	kwa-

1.sg	-mi	1.pl	-(i)thu
2.sg	-kho	2.pl	-(i)nu
Class 1/1a	-khe	Class 2/2a	bo

(3.87) `imoto yami`
IX-car OWN:IX-me
my car

(3.88) `isifo (sa-amntwana >) somntwana`
VII-illness OWN:VII-child
the illness of the child

(We gloss by OWN:IX the suffix which indicates that a thing of class 9 is being owned (the possessee). This suffix is different from POSS:IX, which would mean that a thing of class 9 is the owner.) Zulu shows the canonical structure, where the morphological bracketing is identical from the syntactical one. Different from that are Hungarian and Finnish. In Finnish, the possession may be expressed by the genitive. In addition, there are possessive suffixes to the possessed, which agree in number and person with the possessor. These two are not exclusive. The fact of possession may be expressed as well by the genitive as well as the possessive suffixes. For example,

(3.89) `minun autossa`
me-GEN car-INESS

(3.90) `autossani`
car-INESS-POSS:1.SG

(3.91) `minun autossani`
me-GEN car-INESS-POSS:1.SG
in my car

Notice that the sequence of suffixes is different from the one we would expect. The case suffix is attached before the possessor suffix. In Hungarian, the fact of possession is marked on the possessed noun with certain suffixes. (See Table 3.15. The suffixes have been extremely simplified. All of them may be preceded by a vowel, chosen by vowel harmony, which also determines the shape of `V` in `-tVk`, and the suffix of the 3rd person(s) may be prefixed in addition by `-j-`.) The possessee is in the nominative (which is identical to the bare stem). So, we have the following counterpart of the Finnish examples.

(3.92) `a kocsi-m-ban`
PRT*car* – POSS:1.SG – INESS

Table 3.15: Hungarian Possessive Suffixes

	SG	PL
1ST	-m	-nk
2ND	-d	-tVk
3RD	-(j)a	-k

in my car

(3.93)

(3.94) az én kocsi-m-ban

PRT*me* – NOM*car* – POSS:1.SG – INESS

(3.95) *az én kocsi-ban

PRT*me* – NOM*car* – INESS

Two differences are to be noted. First, the suffixes are in the order possession-case, unlike in Finnish. Second, it is possible to add the emphatic pronoun, but it is not allowed to omit the possessive suffix in that case. This can be explained by assuming that all nouns have case and by assuming that belong assigns nominative case to the possessor in Hungarian. (Notice, by the way, that Hungarian has no genitive.)

Hungarian presents also two other interesting constructions. There are the so-called markers of external possession. The suffix -é denotes the fact that the NP is the owner of something. The latter, however, is not expressed and must be contextually supplied. Therefore, X-é is suitably translated as *the one which belongs to X* or *the one of X*. We gloss it as EPOSS. So, speaking of cars, someone may say,

(3.96) Láttam az édesapá-d-é-t.

see-PAST-1.SG PRT father-POSS:2.SG-EPOSS-ACC

I saw the one of your father.

The external possessor triggers no person agreement, but the number of the external possessor may be marked. The plural of the possessor is marked by a suffix -i

Table 3.16: Hungarian Possessive Pronouns

	SING	PLUR
1.SG	enyém	enyéim, enyémek
2.SG	tied, tiéd	tieid
3.SG	övé	övék
1.PL	mienk, miénk	mieink
2.PL	tietek, tiétek	tieitek
3.PL	övék	övéik

before the possessor agreement suffix, while the plural of the possessee is marked by an `-i` right after the `é` (and before the case ending).

Stem+NUMBER+POSS+EPOSS+NUM+CASE

The first number suffix belongs to the stem; the possessor suffix also expresses the number of the possessor. Therefore, no extra slot for its number is needed.

Finally, there is a series of possessive pronouns that are used predicatively, corresponding to Engllish *mine* in *that car is mine*. These pronouns are listed in Table 3.16. These pronouns are composed from the personal pronouns and the possessive suffixes. For example, `miénk`, is `mi+énk`, roughly translated as *belong of us*. Hence they are parallel to the Sesotho possessive pronouns, which are also a sequence of personal pronoun with possessive prefix, only that the pronoun in Sesotho denotes the possessor and the prefix the possessee.

Prepositions In certain languages, the combination of a preposition and a personal pronoun as a complement can (or must) alternatively be expressed by what looks like an inflectional ending on the preposition. Table 3.17 shows an example of prepositions in Breton, taken from Hemon [46]. Welsh has a similar feature (see Thomas [92]). In Table 3.18 we show the Hungarian prepositions, which behave like the Breton ones. We have added here also two other columns which do *not* correspond to postpositions. Rather, they reflect two locative cases: the inessive and the superessive I (see Chapter 5 for terminology). What is remarkable is that while the inessive is regularly formed with a suffix `-ban` (or `-ben`, depending

Table 3.17: Breton Prepositions

	war on	en in	dre through
1.SG	warnon	ennon	drezon
2.SG	warnout	ennout	drezout
3.SG.MASC	warnañ	ennañ	drezañ
3.SG.FEM	warni	enni	drezi
1.PL	warnomp	ennomp	drezomp
2.PL	warnoc'h	ennoc'h	drezoc'h
3.PL	warno	enno	drezo

Table 3.18: Hungarian Cases and Postpositions

	alatt under	-ben in	-Vn on
1.SG	alattam	bennem	rajtam
2.SG	alattad	benned	rajtad
3.SG	alatta	benne	rajta
1.PL	alattunk	bennünk	rajtunk
2.PL	alattak	bennetek	rajtatok
3.PL	alattuk	bennük	rajtuk

on vowel harmony), the superessive is actually formed by adding -on. However, the forms with the incorporated pronouns use the stem benn- for the inessive, but rajt- for the superessive. It would be a mistake to speak of agreement or inflection in the case of prepositions. What happens here, rather, is that one of the arguments is saturated. It is striking that the borderline is between pronominal arguments and nonpronominal arguments. While the pronominal argument must be realized together with the preposition in the way shown in the tables, a nonpronominal argument not only has to be realized separately (apparently for morphological reasons) but triggers no special form on the preposition. If we had an agreement system, we would expect that the preposition shows 3rd person agreement with its complement if the latter was nonpronominal. This is not observed. Therefore, we must assume that the forms of the Tables 3.17 and 3.18 are combinations P+DP[+pro].

Negation and Complementizers There is also the phenomenon of agreeing complementizers in Flemish and Bavarian. Here is the Bavarian data (see Bayer [8]):

(3.97) `Vater erzähl', wie-st g'schossen hast.`
father, tell-SG.IMP (us) how-2.SG shoot-PERF have-2.SG
Father, tell us how you shot.

The complementizer, `wie` (*how*), also has agreement features like the verb. Notice that the 2nd person pronoun is du, so the data cannot be explained by proposing a phonological reduction of the pronoun. In Finnish, negation is expressed by a negation verb. The verb inflects for person and number. Moreover, it combines with certain complementizers. For example, `jos` (*if*), in a negated sentence can become `jollei` (*if he/she not*). This complementizer inflects regularly.

1.SG	`jollen`
2.SG	`jollet`
3.SG	`jollei`
1.PL	`jollemme`
2.PL	`jollette`
3.PL	`jolleivät`

Both are cases of agreement, even though they function slightly differently. While in Bavarian we can truly speak of the complementizer agreeing with the verb (since the agreement suffixes are doubled), in Finnish the negation verb is actually the carrier of the verbal agreement suffixes.

3.7 The Interaction between Morphology and Syntax

We conclude this chapter with some reflections concerning the interaction between morphology and syntax. This will clarify some questions concerning the exact details of agreement morphology which we had to leave open up till now. The calculus developed so far pairs only meanings with strings of formal elements, which we have called morphemes (or inflectemes). The exact details of how meanings have to be composed were broadly discussed, while we left the matter for morphemes completely open. Here we shall sketch how a genuinely morphological component is added onto the system. We shall assume that in addition to the semantic structures that we have dealt with so far there is another kind of structure. The latter deals with morphological and phonological features of the structure. Merge operates simultaneously on both structures. The morphological structures are similar to the semantic structures. They have a referent system and a 'semantics'. The 'semantics' is the phonological string or a suitable representation thereof. The referent system is a simplified variant of the referent systems of the argument structure. First, no directionality is specified (although that might be done as well). Second, each referent system is allowed to import at most one variable. This variable is therefore not represented at all. It will be seen that it is not necessary at all to represent it, by the rules of the calculus.

The problem of morphology is that the elements are not independent units. Neither is there such a notion of concatenation of morphemes, to which everything can be reduced. For an excellent discussion of the problems of morphology, in particular Latin morphology, see [66]. We shall give an example. The perfect stem in Latin is formed from the present stem by a number of distinct processes. In most conjugation classes, it consists of a *v*, which is inserted after the thematic vowel. So, if the present stem is `laud` ('to praise') with thematic vowel `a`, we get the perfect stem `laudāv`.[11] In the consonantic class, matters are different. Some stems are formed by adding `s`, for example `pinx`, the perfect stem of `ping`, ('to paint'). [12] Others use ablaut, loss of nasalization (which induces lengthening of the vowel), reduplication, and certain mixtures of these processes. The perfect of `tang` ('to touch') is `tetīg` (Ablaut, Denasalization and Reduplication), the perfect of `frang` ('to break') is `frēg` (Ablaut and Denasalization), the perfect of `rump` ('to break') is `rūps` (Denasalization and `s`-Insertion). These changes are unpredictable on the basis of the phonological shape alone. However, each indi-

[11] In this section we shall depart from our practice to consider only written language. We shall not use hyphens to denote incompleteness of a stem or affix. Incompleteness is not signalled in the string itself, rather there will be an end marker (the blank, for example).

[12] Here, the grapheme `x` gets in our way. Phonologically, it is the concatenation of [`k`] and [`s`]. So, the grapheme `x` stands for the phonological string [`ks`]. (This leaves the devoicing of `g` to be explained here. We will not discuss such details, however.) In case of doubt, forms are to be treated as quotes of phonological strings, not of typographical strings.

vidual process (Denasalization, Reduplication, Ablaut) is more or less determined. We must conclude therefore that the different perfect forms are not the result of phonologically conditioned processes, but rather of morphologically conditioned processes. How can this be done? We shall assume that certain sequences of phonemes or some suitable phonological representation thereof is the meaning of the morphological string. So, when we compose two sequence of morphemes, we likewise compose the phonological representations as we did before with the semantics. Take the case of the verbal stem `laud`. We have written `/laud/` to denote the abstract morphological entity which is typically written `laud` (without brackets!). Now we use `[laud]` to denote its phonological representation. To form the perfect stem from the root we must add the thematic vowel and then `v`. This can be done by concatenating three elements:

(3.98) [laud] ⋆ [ā] ⋆ [v] = [laudāv]

How do we know which thematic vowel to insert? We shall simply assume that at the morphological level the elements also have AISs, though in somewhat simplified form. The name space then contains purely morphological information, such as: declension class and thematic vowels. However, we shall not say that the verb selects its thematic vowel, only that it has (in the case of `laud`) the conjugation class `a`, which means that its thematic vowel is `a`. This can be done by assigning to a class of verbs, called *a-verb*, which groups together all verbs with thematic vowel `a`.

(3.99)

$$\begin{array}{|l|}\multicolumn{1}{c}{/\mathtt{laud}^{\checkmark}/}\\ \hline \langle \vartriangle : \begin{bmatrix}\text{MCLASS} & : & \textit{a-verb}\\ \text{THCLASS} & : & \star\end{bmatrix} \rangle \\ \hline [\mathtt{laud}] \\ \hline \end{array}$$

(Notice the absence of variables.) This means in particular that the item has morphological class *a-verb*, which stands for the fact that it is belongs to the vocalic group with thematic vowel `a`, and that the thematic vowel is missing. The entry for the vowel `a`, which we write A-TH, is as follows.

(3.100)

$$\begin{array}{|l|}\multicolumn{1}{c}{/\text{A-TH}/}\\ \hline \langle \Diamond\oslash : \begin{bmatrix}\text{MCLASS} & : & \textit{a-verb}\\ \text{THCLASS} & : & \star \mapsto a\end{bmatrix} \rangle \\ \hline [\mathtt{a}] \\ \hline \end{array}$$

It these two are combined, the corresponding phonological representations are concatenated in the direction specified.

(3.101)

$$\begin{array}{c} /\mathtt{laud}^{\checkmark}/ \\ \begin{array}{|l|}\hline \langle \triangle : \left[\begin{array}{lcl}\text{MCLASS} & : & \textit{a-verb}\\ \text{THCLASS} & : & \star\end{array}\right]\rangle \\ \hline [\mathtt{laud}] \\ \hline\end{array}\end{array}$$

$$\bullet \quad \begin{array}{c} /\text{A-TH}/ \\ \begin{array}{|l|}\hline \langle \Diamond\oslash : \left[\begin{array}{lcl}\text{MCLASS} & : & \textit{a-verb}\\ \text{THCLASS} & : & \star \mapsto a\end{array}\right]\rangle \\ \hline [\mathtt{a}] \\ \hline\end{array}\end{array}$$

$$= \quad \begin{array}{c} /\mathtt{lauda}/ \\ \begin{array}{|l|}\hline \langle \triangle : \left[\begin{array}{lcl}\text{MCLASS} & : & \textit{a-verb}\\ \text{THCLASS} & : & a\end{array}\right]\rangle \\ \hline [\mathtt{lauda}] \\ \hline\end{array}\end{array}$$

(The reader is warned that /lauda/ is merely a way to denote the string of two morphemes. It would be more accurate to write /laud$^{\checkmark}$/$^\frown$/A-TH/. But we have previously allowed ourselves to write /laud$^{\checkmark}$$^\frown$a-th/ to indicate its status as a (higher) unit. Now we even allow ourselves to quote this sequence by its phonological form, which is /lauda/.) Now, the entry for the perfect stem in the vocalic case is as follows.

(3.102)

$$\begin{array}{c} /\text{PERF}/ \\ \begin{array}{|l|}\hline \langle \Diamond : \left[\begin{array}{lcl}\text{MCLASS} & : & \textit{v-verb}\\ \text{THCLASS} & : & a \sqcup e \sqcup i\end{array}\right]\rangle \\ \hline [\mathrm{v}] \\ \hline\end{array}\end{array}$$

(Here, *v-verb* denotes the class of all verbs with vocalic stems.) This says that the perfect suffix combines with any vocalic verb on condition that it has a thematic vowel (of any form) affixed to it. We can combine this suffix with the previous sequence and get

(3.103)

$$\begin{array}{c} /\mathrm{laud}^{\checkmark}{}^\frown\text{A-TH}^\frown\text{PERF}/ \\ \begin{array}{|l|}\hline \langle \triangle : \left[\begin{array}{lcl}\text{MCLASS} & : & \textit{verb}\\ \text{THCLASS} & : & a\end{array}\right]\rangle \\ \hline [\mathrm{laudav}] \\ \hline\end{array}\end{array}$$

Notice that it is the perfect stem that selects the thematic vowel by selecting not the verbal root (without the thematic vowel) but rather the root plus thematic vowel.

We may alternatively consider the perfect stem to come in several allomorphs, one for each vowel class. This however escapes the generalization about the uniformity of this vowel across the tenses. In the consonantic class, however, there is no choice but to posit several allomorphs of the perfect 'suffix'. Each allomorph selects its own class of verbs. The easiest in this respect is the s-Suffix. We shall group all verbs that only take this suffix into a class, and call it *s-verb*. Then the s-allomorph of the perfect stem gets the following representation (with − a value of THCLASS denoting absence of thematic vowel).

(3.104)
$$\begin{array}{c}/\textsc{perf}/\\ \hline \multicolumn{1}{|l|}{\langle\Diamond : \left[\begin{array}{lcl}\textsc{mclass} & : & \textit{s-verb}\\ \textsc{thclass} & : & -\end{array}\right]\rangle}\\ \hline \multicolumn{1}{|l|}{[\mathsf{s}]}\\ \hline\end{array}$$

What will now happen with the perfect in the other cases? Here we must assume that the phonological representation contains something different from a mere string. Namely, we shall introduce abstract functions, such as *abl*, *den* and *red*, which stands for *Ablaut*, *Denasalization* and *Reduplication*, respectively. These functions may be partial, that is to say, they need not be always defined. For example, in the case discussed here the functions are defined only on monosyllabic morphs. We group the verbs simply according to the individual processes that are being applied to form the perfect stem. For example, ag$^{\checkmark}$ ('to drive') is a member of the ablaut-class. Its perfect is just formed by means of ablaut.

(3.105)
$$\begin{array}{c}/\mathsf{ag}^{\checkmark}/\\ \hline \multicolumn{1}{|l|}{\langle\Diamond : \left[\begin{array}{lcl}\textsc{mclass} & : & \textit{verb}\\ \textsc{thclass} & : & -\\ \textsc{abl} & : & +\end{array}\right]\rangle}\\ \hline \multicolumn{1}{|l|}{[\mathsf{ag}]}\\ \hline\end{array}$$

Hence, there is an ablaut allomorph of PERF that is as follows.

(3.106)
$$\begin{array}{c}/\textsc{perf}/\\ \hline \multicolumn{1}{|l|}{\langle\Diamond : \left[\begin{array}{lcl}\textsc{mclass} & : & \textit{verb}\\ \textsc{thclass} & : & -\\ \textsc{abl} & : & + \mapsto -\end{array}\right]\rangle}\\ \hline \multicolumn{1}{|l|}{\textit{abl}}\\ \hline\end{array}$$

Note that there is nothing wrong with PERF having several representations. These correspond to its various allomorphs. This is just the same as and having different semantic representations in 2.9. The convention will now be the following. If the functor (in this case the perfect stem) carries a function in place of a string, then this function is applied to the argument. Consequently, we get

(3.107)

$$
\begin{array}{c} /\mathtt{ag}^{\checkmark}/ \\ \begin{array}{|l|}\hline \langle\Diamond : \left[\begin{array}{lcl} \textsc{mclass} & : & \mathit{verb} \\ \textsc{thclass} & : & - \\ \textsc{abl} & : & + \end{array}\right]\rangle \\ \hline \mathtt{[ag]} \\ \hline \end{array}\end{array}
\bullet
\begin{array}{c} /\textsc{perf}/ \\ \begin{array}{|l|}\hline \langle\Diamond : \left[\begin{array}{lcl} \textsc{mclass} & : & \mathit{verb} \\ \textsc{thclass} & : & - \\ \textsc{abl} & : & + \mapsto - \end{array}\right]\rangle \\ \hline \mathit{abl} \\ \hline \end{array}\end{array}
$$

$$
= \begin{array}{c} /\mathtt{ag}^\frown\textsc{perf}/ \\ \begin{array}{|l|}\hline \langle\Diamond : \left[\begin{array}{lcl} \textsc{mclass} & : & \mathit{verb} \\ \textsc{thclass} & : & - \\ \textsc{abl} & : & - \end{array}\right]\rangle \\ \hline \mathit{abl}(\mathtt{[ag]}) \\ \hline \end{array}\end{array}
$$

Since *abl*([ag]) = [eg], we get the desired result. (Notice that there is also vowel lengthening involved, but we will ignore that fact here.) This will allow to let certain restrictions on combinations be a corollary of the fact that these are only partial functions. However, this also raises delicate questions. If there are complex verbs, say `corrumpere` (= `con+rumpere`) then we expect that the function *den* must be applied to the root before the prefix is added. This has the advantage that the perfect of the complex verbs is easily predictable from that of the root. Since we have the perfect stem `rūps` for `rump`, we must have `corrūps` for `corrump`.[13] Now, in what form must this complex verb be represented in the lexicon if this is the case? Presumably we shall have to say that it is stored in two parts, namely in the form prefix+root. That this is not such a strange idea is shown by German verbs. The perfect is a combination of a suffix, which is typically `t`, and a root prefix `ge`, which is put in between the prefix and the verbal root (for example we have `aus-ge-lach-t`, from `aus-lachen` ('to laugh at'). This happens only if the verb is segmentable. If it is nonsegmentable the suffix appears alone and the root prefix is not added (`zer-legt` from `zer-legen` ('to take apart, dissect'). Segmentable verbs are those verbs that leave the prefix behind when being moved

[13]There is only one exception to this rule. If a root undergoes reduplication, this is generally *not* the case with the combination of prefix and root. So, we have `tetigī` but `at+tigī` and not `at+tetigī`, which is to be expected from the present form `at+tingere`. So the prefix 'swallows' the first syllable of the reduplicated root.

to 2nd position while nonsegmentable verbs must move as a whole.

(3.108) `Daniel lachte seine Kollegen aus.`
Daniel laughed at his colleagues.

(3.109) `Daniel zerlegte sein Motorrad.`
Daniel took apart his motorbike.

This illustrates that the words are stored in the lexicon with explicit information about their segmentation and segmentability. Certain phonological and morphological processes are sensitive to this segmentation.

On the other hand, we get ablaut in certain complex verbs in Latin, for example `attingere` (= `ad+tangere`). The representation of this verb must then look like: $[\mathsf{ad}] \star \mathit{abl}([\mathsf{tang}])$.

This theory of morphophonological representations takes care of allomorphy. It allows us to postulate distinct shapes for some morpheme and specify the environments in which they appear, be they semantic or morphological, or both. Of course, some phenomena need not be dealt with in this system, namely all those which are of truly morphophonological character. This appplies to vowel harmony in Finnish. Many suffixes come in two shapes, with the vowel either back or front. For example, the inessive has the forms `ssa` or `ssä`. The rule is as follows. If the word to which it is attached consists of at least one back vowel and no front vowels, then the suffix `ssa` is taken (`talo-ssa`, in the house). Here, `a`, `o` and `u` count as back, `ä`, `ö` and `y` count as front, and `e` and `i` as neutral. If the word consists entirely of back or neutral vowels, then the suffix `ssä` is taken (`hissi-ssä`). This phenomenon can be dealt with in purely phonological terms, if we assume that the lexicon specifies the suffix in the following way: its form is given as `ssA`, where `A` abbreviates the set $\{\mathsf{a}, \mathsf{ä}\}$. If shipped to the phonology, the latter will then insert the correct vowel on the basis of the just mentioned criteria. However, we may additionally specify each root and each suffix as $\pm$*back* and state that only likes can combine, a +*back* root with a +*back* suffix, and a −*back* root with a −*back*. These two strategies are not even exclusive. The latter of course misses the generalization concerning the vowel harmony.

Next we turn to the issue of unrealizable units. The ending `ris` in Latin signals 2nd person singular passive, while `mini` signals 2nd person plural passive. These are the passive allomorphs of `s` and `tis`. On a phonological basis there is no segmentation of the two suffixes into units U, V and W such that

$$(3.110)\quad \begin{array}{ccccc} \mathsf{U} & \star & \mathsf{W} & = & [\mathsf{ris}] \\ \mathsf{V} & \star & \mathsf{W} & = & [\mathsf{mini}] \end{array}$$

We conclude that the element '2nd' and 'singular' are not pronounceable by themselves. If one investigates the full paradigm this is corroborated. But clearly these are semantic units and from a morphological point of view nothing speaks against treating them as units. It is only the problem of phonological realization that stands in the way. There are at least two possible scenarios. The first is to add to the phonological layer some abstract elements, here U, V and W, together with the properties above. This is mathematically sound but gives rise to entities that have no phonological basis. Alternatively, we may introduce abstract functions, f_U, f_V, f_W and the equations

$$(3.111)\quad \begin{array}{lcl} f_W(f_V(\mathsf{X})) & = & \mathsf{X} \star [\mathsf{ris}] \\ f_W(f_U(\mathsf{X})) & = & \mathsf{X} \star [\mathsf{mini}] \end{array}$$

However, also this is unsatisfying to some degree since the action of the functions cannot be spelled out individually.

The system now takes the following shape. A sequence of elements acts on two different tiers: the syntactic tier (which is what we display most of the time) and on the morpological tier. A full description of the lexical entry for the word $\mathsf{laud}^{\checkmark}$ is as follows.

(3.112)

$$\begin{array}{|l|} \hline \langle e : \vartriangle \bigcirc : \star \rangle, \\ \langle x : \triangledown \text{⦶} : [\text{CASE} : \mathit{nom}] \rangle \\ \langle y : \triangledown \text{⦶} : [\text{CASE} : \mathit{acc}] \rangle \\ \hline e \\ \hline \mathsf{praise}'(e); \mathsf{act}'(e) \doteq x; \\ \mathsf{thm}`(e) \doteq y. \\ \hline \end{array} \quad \longleftarrow \quad \text{semantic-syntax tier}$$

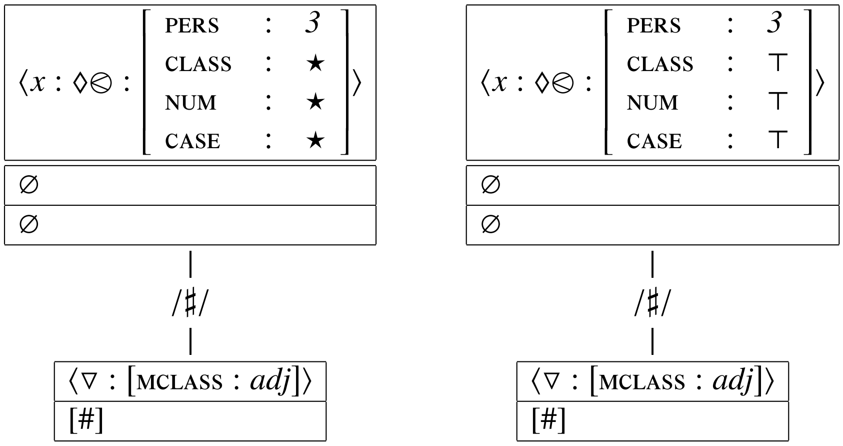
$$|$$
$$/\mathsf{laud}^{\checkmark}/$$
$$|$$

$$\begin{array}{|l|} \hline \langle \vartriangle : \begin{bmatrix} \text{MCLASS} & : & \mathit{a\text{-}verb} \\ \text{THCLASS} & : & \star \end{bmatrix} \rangle \\ \hline [\mathsf{laud}] \\ \hline \end{array} \quad \longleftarrow \quad \text{morphology-phonology tier}$$

We have called the first tier the **semantic-syntax tier** (**SX-tier**) and the lower one the **morphology-phonology tier** (**MP-tier**). When we merge two units, the merge is carried out in both tiers. By consequence, conditions on merge apply in parallel. It can succeed only if the conditions of the SX-tier and those of the MP-tier are being met. For example, the verb `laudare` selects two arguments, one being a nominative marked NP and the other an accusative marked NP. However, the merge will not succeed with laud$^{\checkmark}$, since the latter is a root and not a complete word. This is an issue to look at next. We mention only a few useful details. We shall allow two units to merge in case they can merge at one tier alone, and no conditions exist at the other. For we assume that MP-merge can succeed if the argument structures are empty. Then we simply concatenate.

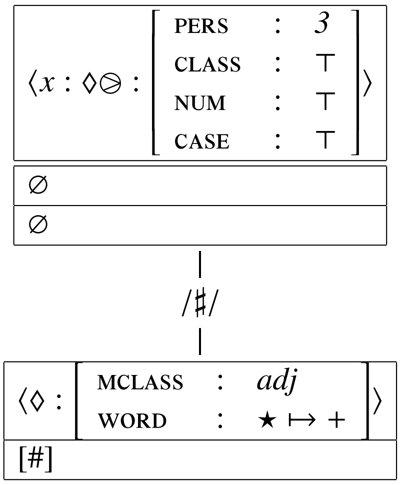
We have at several occasions said that certain affixes are optional, and that it is not necessary that all values of features be instantiated before we have a complete word. What we have not spoken about is where it is decided that we have a word. This is a delicate matter. Since the affix is the functor, we cannot write into the root when it is a complete word, since the root cannot choose with which functors it will combine. Only the functor decides which arguments it chooses. Therefore, we choose a different solution. We shall propose that the word boundary is a functor that selects to its left a morphological unit and turns it into a word. It is the word boundary into which is written what is a word and what is not. For example, the word boundary marker for Hungarian will state that no adjective may carry inflection, while in Latin all adjectives do. Figure 3.12 shows these two lexical entries. Thus, what the word boundary marker does is to remove the entry from the morphological tier. Now syntax is alone responsible for the merge. Notice that we allow merge in the morphology when the referent systems are empty. Then we simply concatenate. There are however occasions where we want to keep morphology responsible. Therefore we allow the boundary marker to simply mark the argument by the feature [WORD : +], to state that it is a syntactic

Figure 3.12: Word Boundary Marker for Hungarian (left) and Latin (right) Adjectives



word.

(3.113)



At first sight it might be thought that this is more complicated than necessary. But there are a number of phenomena which motivate this proposal. Lets look again at raising verbs of Section 2.8. We have said there that raising verbs in addition to fusing with their verbal complement require it to be a word. To avoid confusion we shall call this a 'syntactic word'. Thus, whether or not something is a syntactic word is independent of the question of whether it is a word in the ordinary sense of the word. The complex `laten zwemmen` is a syntactic word, but contains two

Figure 3.13: Dutch `laten`

$\langle e : \vartriangle \varnothing : \textsc{inf}\rangle$,
$\langle y : \triangledown \oslash : \textsc{acc}\rangle$, $\langle f : \blacktriangledown \ominus : \textsc{inf}\rangle$.
e, f
let′(e); act′$(e) \doteq x$;
thm′$(e) \doteq f$; ben′$(e) \doteq y$;
act′$(f) \doteq y$.

/laten/

$\langle \lozenge : [\textsc{word} : +]\rangle$
[laten]

words. We shall say that what the boundary marker does in this case is to create a word in the intuitive sense. This it does by attaching a certain break (denoted by #) at the end of the phonological sequence. However, it also sets the value of the feature WORD from $\star$ to $+$. We shall assume that this feature tells us whether or not we have a syntactic word. To see what this is good for, let us look at fusion again. We shall assume that fusion happens if and only if the merge operates on both tiers simultaneously. Further, we have assumed that morphology, in contrast to syntax, allows only one argument per unit. This means—more or less—that morphology is regular (as a formal language). But it also means that there can be at most one argument in the SX-tier which is fused. The last assumption to be made is that morphological merge precedes syntactic merge. So, this means that if there is an argument that is being fused, then it must be discharged first. This generates the same set of requirements as those we have made in Section 2.8. Now let us look at the complete entry for `laten`. It is shown in Figure 3.13. The last entry is the fused f. By our conventions, the first merge will identify f, and so we know that when we merge with some element, say `zwemmen`, then both the SX-tier and the MP-tier must merge. At the MP-tier we find that the complement looked for must be a syntactic word, and that the result will again be a syntactic word. Actually, if all this is assumed, the distinction between $\blacktriangledown$ and $\triangledown$ and between $\blacklozenge$ and $\lozenge$ lies exclusively in whether the verb is a morphological functor or not. Hence, we can in principle dispense with that notation. However, it can be instrumentalized in a different way.

Fusion interacts with morphological properties of particles in a complicated way in this system to create the distinction between what we know as *word affixes* and *phrasal affixes*. Now, clearly, most morphological affixes must be fusional, otherwise they can never be attached to a word that needs an argument. But there are affixes that are nonfusional. For example, case markers of Hungarian, Turkish and Japanese are such elements that require a phrase as a complement. Technically, they are adjuncts, but they are adjuncts that do no not take unsaturated argument structures. So, we eventually say the following. Morphological arguments must be the first to be discharged, hence they appear at the end of the argument structure. By our conventions, it is always clear whether or not an argument is morphological or not, we just have to look at the MP-tier. Additionally, we can use the $\triangledown/\blacktriangledown$ and the $\lozenge/\blacklozenge$ distinction to mark whether the element is fusional or not. Fusional arguments are obligatorily morphological.

So, we have elements that can be affixed only to a word, while others can be affixed only to an entire phrase. We shall meet this distinction in the Chapters 4 and 5, where we shall show that cases are phrasal affixes in some languages and word affixes in others. However, since a single word can be a phrase, it would be a mistake to conclude that we only have to say that certain elements can only be affixed to words and others only to complex units. Moreover, not every complex unit is a phrase. In fact, the notion of a phrase is a derived notion of the argument structure. The notion of a word in the context of affixes is not the syntactic word but the word in the ordinary sense. This creates something of a dilemma: if the Hungarian cases are affixes (so they are suffixes of some word) but need a phrase to combine with, how can it satisfy both restrictions? The answer lies in the distinction between phonological and morphological restrictions. A suffix simply cannot attach to a word with a boundary marker for phonological reasons. This distinguishes it from a clitic. However, as long as the boundary marker is not present, the string may be as heavy as it possibly can. Thus the fact that the affix is phrasal affects the syntactic nature of the string that it combines with (it must be a phrase), the fact that it is a suffix of some word affects the fact that it cannot combine with a boundary marker. Notice that the word boundary marker marks only the right edge never the left edge.

So, we end up with the following classification: a word affix is an affix that can only be combined with words; a phrasal affix is an affix is allowed to combine with a word or a sequence of words, provided that they form a phrase. Notice that this classification allows for a third kind of affix, namely one that is not a word affix but may be combined with any type of argument structure, not necessarily phrasal. One naturally wonders if such elements exist. But consider a plural markers that is of this type. It would simply be allowed to appear at any place in the NP. Such markers have been claimed to exist, for example in [?].

Notes on this section. As the reader may have noted, the mechanism for spelling out a concrete word is actually nonlinear. Otherwise, an account of denasalization or of in- or circumfixes is difficult to imagine. However, we need not assume too much power here. All we need is to assume that the representation contains in place of a single string a (bounded) sequence of them. In the cases above, two strings are enough. For Arabic, more is required. Roots consist of generally no more than three consonants (for example `ktb` ('to write')), and morphology adds vowels in between them or some material preceding or following it (`katab` (perfect active), `uktab` (imperfect passive) etc., see [88]). There is no problem in handling even Arabic here. Just assume that the root is a sequence of three consonants, each of which can be manipulated individually. Note that in syntax there are also phenomena of the sort just discussed. One example is the verb second phenomenon in German, which splits the verb and its prefix, or perhaps even negation in French. In doing so, we are actually assimilating what is generally believed to be a syntactic phenomenon to a morphological one. Exactly this has been advocated by Anderson [4], following an original suggestion by Jakob Wackernagel.

Chapter 4

Parameter

In this chapter we shall introduce *parameter* into the referent systems. The mechanics of parameter is distinct from that of ordinary variables. Parameter are the kind of variables that are always present even when they are not needed. Prototypical examples are time points. Many nouns are sensitive to time points, while many are not. However, unlike verbs, the dependency on time has no (or next to no) syntactic or morphological relevance. The omnipresence of parameter offers the possibility to let structures choose freely their set of parameter on which they depend without changing their combinatorial possibilities.

4.1 Properties

In this chapter we shall introduce a new construct, that of *parameter*. The first example of a parameter are *properties*. The behaviour of parameter will be motivated through properties. The introduction of parameter will have repercussions for several basic structures of which we shall discuss mainly two: the NP and the tense complex.

Properties are a sort we have not discussed before. There are mainly three reasons for introducing properties. One is that there are plenty of facts showing that properties are distinct from objects or any other kind of entity that we have introduced so far. The other reason is that the mechanism for the assignment of meaning to inflectional morphemes has various technical disadvantages that can only be solved if we assume the existence of properties. The third reason is that because we have no direct mechanism for abstraction, if we want to form properties in the semantics we must actually assume that they exist beforehand. We shall address these questions in turn.

We have seen so far that there are objects, coming as entities and groups, and that there are events. But there is plenty of evidence that we must assume also properties. Here are some constructions in which an NP or an adjective actually denotes a property.

(4.1) `John is a wizard.`

(4.2) `John is clever.`

(4.3) `People call John a fool.`

(4.4) `Paul eats the meat raw.`

In (4.1), the property of being a wizard is attributed to John. In (4.2) it is the property of being clever. (4.3) says that the people think that John has the property of being a fool. Finally, (4.4) says that Paul is eating the meat and the meat is in a raw state, that is, having the property of rawness. This motivates the addition of properties into our ontology. However, we must always ask whether the addition of a new sort it really necessary. Perhaps it is possible to construe these examples nevertheless with objects in place of properties. For example, we might say that (4.1) equates John with a person who is a wizard. Perhaps in this example this is feasible. Notice however that no equational reading of that sentence is available in the same sense as for example (4.5) and (4.6) where we refer to an already established individual.

(4.5) `John is the wizard.`

(4.6) `John is one of the wizards.`

In both examples there is an individual or group introduced in postcopular position and it is said that John is that individual ((4.5)) or is part of the group ((4.6)). We claim that no such reading exists for (4.1). This sentence simply attributes a property to John. The same holds for the other constructions.

One difference between objects and properties is that objects can be used with a demonstrative while properties cannot.

(4.7) *`They call John this fool.`

This leaves us with the picture that NPs may alternatively denote properties and objects. This is indeed the case. Evidence for this view comes from Hungarian. As [33] argue, Hungarian shows incorporation.

(4.8) `Az orvos a beteget vizsgált.`
The doctor examined the patient.

(4.9) `Az orvos beteget vizsgált.`
The doctor examined patients.

The difference between (4.8) and (4.9) is among other things the unavailability in (4.9) of backward anaphoric reference to the object. While Farkas and de Swart analyse this as a lack of discourse marker, we propose here to view this as a property denoting phrase (notice the absence of the determiner). In the present system this will introduce the possibility of blocking reference to the kind if necessary.

From a technical point of view it is absolutely necessary to have properties, be they sorts in their own right (as in property theory) or simply individual concepts. Otherwise the semantics of non-intersective adjectives must remain mysterious. Non-intersective adjectives are such adjectives that modify a property rather than attributing a property to an object. An example is *good*. A good teacher might not be a good person, since he or she might just be good at teaching. Therefore, in order to say that someone is a good teacher it is not to say that he is good and a teacher, rather, he is good at teaching. Similarly *big* or *tall*. A big mouse is not of the same size as a big elephant. In order to know whether something is big you need to know in what respects it is big. Something can be a small mammal but a big mouse. In these examples it is patently clear that the adjective cannot simply take the object and attribute a property to it, as we assumed previously. Rather, the adjective must know what property was said to hold of that object. Therefore, the property must be explicitly represented. The same holds by the way also for the plural, but we shall get to that anyhow later.

The second reason we shall adduce here in defense of properties is the problem of the association of meaning to morphemes. So far we have assumed that each and every morpheme has a meaning. This applies, for example, to plural. So whenever the plural morpheme is attached to some stem it forms a group of things satisfying that property. But exactly this cannot work. Consider the following example from Latin:

(4.10) `quattuor magni mures`
four big-MASC.PL.NOM mouse-PL.NOM
four big mice

According to our previous analysis, this would be a group of mice which is in addition a group of big things and a group consisting of four elements. But we cannot construe the adjective like that. Without knowing what property is considered we cannot know whether the right kind of group is formed. The mice, being mammals, are small mammals. So any group of four mice is a group of four mice which are small mammals. But not every group of four mice which are small mammals will qualify for a group of four big mice. It is therefore useless to ask whether the group consists of big things. Given the group, it may consist of big things when looked at it from one perspective (mice), and of small things when looked at it from another side (mammals). Hence, our previous proposal is doomed to failure with respect to non-intersective adjectives.

What can be done? We shall assume that the numeral, in this case `quattuor`, actually forms the group. Before it does so there is simply no group, just a property. So, we consider both nouns and adjectives as denoting properties. (Sometimes even noun phrases denote properties as we have seen in example (4.1).) We shall take it that the numeral forms the group. This has an immediate consequence. Namely, after the group has been formed it is opaque for non-intersective adjectives. This seems to be incorrect, but we shall hold onto it for the moment. Consider by way of counterexample the sentences (4.11) and (4.12).

(4.11) `This teacher is good.`

(4.12) `This mouse is big.`

(4.11) says that a certain individual is good as far as his abilities as a teacher are concerned. (4.12) says that the individual is big for a mouse. We shall leave the matter at that, however.

There is a third point that deserves mentioning. We shall assume throughout that there is no mechanism for abstraction. Consequently, there is no way to obtain a property from an individual or a group. Thus, if we assume that a noun, say *mouse*, only denotes certain objects there is no way have it denote a property.

To get this property one needs abstraction. Recall namely that the word `mouse` has been given the meaning $\mathsf{mouse}'(x)$, which is an open formula. Now, in order to obtain a property from that formula we need internal devices to abstract over a variable. We shall assume, however, that there is no such mechanism. The reason is twofold. First, we assume that language does in fact not use abstraction (at least not in the form of λ-calculus), but prefers to talk rather concretely, that is, using objects whenever possible. The second is that we do not wish to introduce λ-calculus through the back door, since that would make the system rather costly (in terms of processing requirements). More on that in Section **??**.

4.2 The Mechanics of Parameters

The reason that we have not introduced properties earlier is that their behaviour is rather complex. Intuitively, properties are not things, and language likes to talk of things and in terms of things. Or, to put it differently, the argument structure consists in a classificatory system that is mainly geared towards things and is therefore rather ill-adapted to handle abstract entities. As a consequence, the agreement mechanism, primarily invented to classify concrete objects, is used for all kinds of denotations, be they groups or abstract entities like properties. One can observe for example that the gender system of Indo-European is an obscure mixture of semantics and morphology. The motivating terminology is the distinction between animate and inanimate, and between male and female among the animate. By definition, any abstract entity must be neuter. However, this is very often not the case. Instead, the system is arbitrarily extended.

Moreover, what we find is that independently of what the adjective actually denotes, the agreement system targets the noun phrase denotations, if anything. For it is at the level of noun phrase where the expression actually an object; before that it simply denotes a property of some kind. In order to account for that we shall assume that properties are actually not classified independently; rather, a property is assigned a variable of an object whose classification features it uses. Whether or not that object variable actually occurs in the semantics will be irrelevant. One may think of this object variable as an object that is in the process of being made. Moreover, there is a fundamental difference between objects and properties. Namely, the meaning of an adjective is usually not a property but rather a function modifying a property. An adjective consumes a property, say q, and returns another property, p. For example, the meaning of `big` is $\lambda q.\lambda x.\mathsf{big}'(q)(x)$, where q is a variable for a property. This means that it asks for a property (here q) and an individual, x, and attributes a property to x, namely, the property of being big with respect to being q, that is, being a big q-er. So, adjectives modify the property that is attributed to the object. This is in stark contrast to the way the system was assumed to work. The agreement within the noun phrase was made possible through the coherence of the objects that are being used within the argument structure. Since the noun and all the other adjectives were attributes of the same object they showed agreement by virtue of being predicates of the same object. When the object is gone, the coherence is lost. We shall have to look for it elsewhere. The idea that saves us from loosing coherence is the notion of a *parameter*.

Before we start to develop the semantical representation that handles the adjectives we shall say that the idea of certain things changing through the structure is actually quite pervasive. For example, time is constantly being reset, not only from one sentence to another. Properties also depend on time; for example, being a prime minister or a director is a time dependent property and language has means to keep track of the time at which a property applies to which object. Similarly, worlds or situations can be reset. When we talk about fictitious things it is not assumed that they exist in this world. Again, there are controlled ways to track the current value of worlds or situations. Last but not least the coordinates speaker/hearer can be reset in a text. We call all these things *parameters*. (In the literature they are also referred to as indices.) The idea that we shall develop is that while hand shake of referent systems is brought about by sharing an object variable, this handshake can also bring about a sharing of parameters. In order to do this, the parameter is associated with a particular variable. When the variable is shared, so is the parameter associated with that variable. To see how this works we shall outline the semantics of an adjective. This means that we shall study the mechanism of a single parameter. Later we shall be concerned with additional parameters. We shall annotate the name of the referent with a letter, choosing p, p' and q for properties. The parameter is separated from the name (or name change statement) by a double colon (::).

(4.13)

`/big/`
$\langle x : \Diamond\ominus : \nu :: p \mapsto p' \rangle$
$\varnothing$
$p' \doteq \mathsf{big}'(p).$

So, the parameter of a property is added after the name. Notice that the name may change as well as the parameter. Since the parameter is associated with the name, the value of a parameter can only be reset through passing on the object. This can be seen with a noun. Nouns do not modify a property, hence they only instantiate the parameter. The lexical entry for a nonrelational noun will now take the following shape.

(4.14)

$$\begin{array}{|c|}\multicolumn{1}{c}{\texttt{/mouse/}}\\ \hline \langle x : \triangle\,\bigcirc : \begin{bmatrix}\textsc{pers} & : & 3\\ \textsc{class} & : & \mathit{neut}\\ \textsc{num} & : & \star\end{bmatrix} :: p\rangle \\ \hline \varnothing \\ \hline p \doteq \mathsf{mouse}'. \\ \hline \end{array}$$

So, a noun does no longer denote an entity. It now denotes a property. As before we shall assume that x comes out of the lexicon with certain features being instantiated. It is a morphological requirement to fill some of the remaining features by means of inflectional morphemes.

So, how is now that two structures like the above merge? We shall assume that the merger functions as before with respect to all the nonparametrical stuff. In particular, the variable x is identified by the adjective to its right under the name ν. When it combines with a noun with variable x that has the name ν then the merge succeeds, and x of the adjective and x of the noun become shared. Now, x in the adjective has a property parameter p and x is the noun has a property parameter p. In virtue of the variables being shared, the parameters will be shared as well. So, as a result we get

(4.15)

$$\begin{array}{|c|}\multicolumn{1}{c}{\texttt{/big mouse/}}\\ \hline \langle x : \triangle\,\bigcirc : \begin{bmatrix}\textsc{pers} & : & \mathit{3}\\ \textsc{class} & : & \mathit{neut}\\ \textsc{num} & : & \mathit{sing}\end{bmatrix} :: p'\rangle \\ \hline \varnothing \\ \hline p \doteq \mathsf{mouse}';\\ p' \doteq \mathsf{big}'(p). \\ \hline \end{array}$$

The parameter p is being shared resulting in the following semantics: `big mouse` denotes a property of being a big p-er, where p is the property of being a mouse. One can eliminate the occurrence of p as it has become technically irrelevant.

What we get is

(4.16) $p' \doteq \mathsf{big}'(\mathsf{mouse}')$

When there are several parameters, it must be made clear what kinds of parameters there are, and it must be assumed that there is of every kind only one parameter. Those parameters that are not explicitly mentioned but are provided by the argument will be passed on unchanged.

The semantics of intersective adjectives was previously straightforward but has been complicated, since intersective or not, an adjective needs to keep track of the property parameter. Given that red' is a property of individual (and not a property of properties), the adjective `red` has now the following semantical structure.

(4.17)

/red/
$\langle x : \Diamond\ominus : \nu :: p \mapsto \nu :: p' \rangle$
$\varnothing$
$p' \doteq \lambda x.\mathsf{red}'(x) \wedge p(x)$

Notice that there is a fair number of adjectives that are used both non-intersectively and intersectively. An example is `big`. On the one hand, whether something is big or not depends on what kind of object it is, on the other hand there is also an absolute notion of what a big object is. This may affect the range of syntactic constructions in which an adjective can appear. Typically, when used in postcopular position an adjective either has to be intersective or a property must be inferred.

(4.18) `This mouse is brown.`

(4.19) `This mouse is big.`

(4.20) `It is big.`

In (4.18) we may say that the object under consideration, a specific mouse, is brown. Assuming that `brown` is an intersective adjective, this is the most unproblematic usage of the adjective. In (4.19) we are left with two choices. We may consider the adjective `big` as being used in an absolute sense, in which case we really have a really big mouse being talked about, or it is used not in an absolute sense, and then a property must be inferred from the context. Presumably in this example the object under consideration is big in its property of being a mouse. Notice that the adjective cannot be used non-absolutely in (4.20) unless the property in question is contextually given.

Now we shall turn to other parameters. The most pervasive parameter is *time*. Many words are time dependent in one or the other way. Someone is at some point a prime minister, and a revolutionary at another. Something is red at some moment and green at another. So, in addition to properties we shall also have to consider time points as parameters. Our proposal here will not be very deep, and to a large extent reminiscent of what is generally known as Reichenbach's theory of the tenses, though the details may differ. We shall employ three time points (or intervals, to be exact). The first is the *reference time*. By default it is equated with the utterance time. (Comrie [18] uses the term *deictic centre*.) The second is the story time, and the third is the *event time*, or as we shall call it, the *predication time*. By default, event time and story time coincide. The relation between story time and reference time is marked by the simple tenses

- past tense if the story time is prior to the reference time
- present tense if the story time includes the reference time
- future tense if the story time is after the reference time

Complex tenses are used when the event time is distinct from the story time. For example, if the event time is anterior to the story time, then the pluperfect is used. This schema offers more distinctions than are usually made in language, and certainly many other distinctions may be found. We shall not deal here at all with aspect, for example. Now, when we have a simple sentence, each of the time sensitive elements may actually hook onto a different parameter. We shall illustrate this with some German examples. Consider the following sentence.

(4.21) `Der Präsident war in seiner Schulzeit ein schlechter Schüler.`
In his school days the president was a bad student.

Here, the nominal `Präsident`, as it is time dependent, must hook itself onto some parameter. But which one? As story time and event time coincide, there is only a choice between reference time (that is, now) and story time. The preferred reading is when the subject is formed at reference time. In German this can be made explicit by using the adjective `heutig` (*present day*).

(4.22) `Der heutige Präsident war in seiner Schulzeit ein schlechter Schüler.`
The present day president was in his school days a bad student.

However, it is not necessary that the subject be formed at reference time, it can also be formed at event time, see (4.23). The same holds for the object.

(4.23) `Im Jahre 1953 hielt der Präsident eine große Rede.`
In the year 1953 the president held a big speech.

(4.24) `Der Präsident lernte den Minister während seiner Schulzeit kennen.`
The president got to know the minister during his school days.

In (4.23) the reference time is now (hence past tense), but the subject is preferably formed at story time. In (4.24) either interpretation for the object noun phrase are OK. We may either conceive of the minister as being the one at story time or the one at event time. (We assume here that the pronoun refers back to the president, otherwise the preferences are inverted.) We can disambiguate the sentence by using either `heutig` or `damalig` ('of that time').

(4.25) `Der Präsident lernte den heutigen Minister während seiner Schulzeit kennen.`
The president got to know the present day minister during his school days.

(4.26) `Der Präsident lernte den damaligen Minister während seiner Schulzeit kennen.`
The president got to know the minister of that time during his school days.

In (4.25), it is the minister at utterance time (= now) that the president got to know during his school days, while in (4.26) is the minister of story time (= then).

These facts can be accounted for in various ways, and our proposal has only tentative nature. A deeper analysis will have to be given. At this point we shall only show how such an analysis might go. First, when talking about time as a parameter, we shall basically assume that all elements share these parameters. If they do not make use of them that will be fine, but they will still pass them on to all other elements. This means that at all levels we shall have to distinguish three time points (or intervals), namely reference time, story time and event or predication time. This applies equally to nouns and noun phrases. However, basically the noun phrase needs only one time point. We may now say that this time point simply is the predication time of the noun, and that the noun phrase may decide to pass on this point of time either as the reference time or the story time. If this is the analysis, then the NP acts by shifting the the predication time. Another analysis is that the NP does not change the assignment of the parameters but only taps either of them. The disadvantage of the latter analysis is that before a decision is made as to which time points serves for the formation of the NP we must keep the time parameter distinct. We then end up with four parameters rather than three. This is unsatisfactory. We shall therefore assume the first analysis, where the NP is shifting the predication time. To see how this works, we shall give the semantics for nouns and adjectives. The lexical entry for a time dependent noun is like this:

(4.27)

$$\begin{array}{|c|}\multicolumn{1}{c}{\texttt{/Präsident/}}\\ \hline \langle x : \triangle\ \bigcirc : \begin{bmatrix}\textsc{pers} & : & 3\\ \textsc{class} & : & masc\\ \textsc{num} & : & \star\end{bmatrix} :: \begin{bmatrix}\textsc{prop} & : & p\\ \textsc{pred} & : & t\end{bmatrix}\rangle \\ \hline \varnothing \\ \hline p \doteq \mathsf{president}'(t) \\ \hline \end{array}$$

We have used the same attribute value notation for the parameters. Here PRED is the attribute for the predication time. The values are typically variables, in this case t. The constant president′ is a function from time point to properties of individuals. Likewise, an adjective can be time dependent:

(4.28)

$$\begin{array}{|c|}\multicolumn{1}{c}{\texttt{/groß/}}\\ \hline \langle x : \Diamond\ominus : [\textsc{pers} : 3] :: \begin{bmatrix}\textsc{prop} & : & p \mapsto p'\\ \textsc{pred} & : & t\end{bmatrix}\rangle \\ \hline \varnothing \\ \hline p' \doteq \mathsf{big}'(t)(p) \\ \hline \end{array}$$

Before we continue the implementation of time parameters we have to define the mechanics of merge in presence of parameters. An AIS is now expanded by a parameter handling statement.

Definition 4.1 *Let P be a set of* ***parameter names****. A* ***parameter handling statement*** *(PHS) over P is either an AVS with attributes drawn from P and values from the set of referents; we call such types* ***simplex****. Or it is an AVS with attributes drawn from P and values being pairs of referents. We call such PHSs* ***duplex****.*

We write $[P : x \mapsto y]$ in place of $P : \langle x, y\rangle]$. PHSs are quite different from AVSs: the values of attributes (here called parameter names) are referents, and there are infinitely many of them. Only the set of parameter names is assumed to be finite. Even though we speak of 'parameters' and 'referents' there is no difference between the two. The variables are allowed to change from being used in the PHS to being the variable of an AIS and back. For each sort of variable (thing, person, time, world, location) we will actually have several distinct parameter names for variables of that sort (as we did just now for time), but there seems to be an upper bound of four for each.

It is not required that the values for the parameter names are distinct. A referent can appear in as many places as it likes.

Definition 4.2 *A* ***parametrised argument identification statement*** *(PAIS) is a pair consisting of an AIS* $\langle x : \partial : N\rangle$ *and a PHS* $\mathcal{P}$*, written* $\langle x : \partial : N :: \mathcal{P}\rangle$*, such that (a) if* $\partial = -\bigcirc$ *then* $\mathcal{P}$ *is empty, (b) if* $\partial = \vartriangle \flat$ *or* $\vartriangle \bigcirc$ *then* $\mathcal{P}$ *is simplex, and (c) if* $\partial = \lozenge\flat$ *then* $\mathcal{P}$ *is duplex. We say that* $\langle x : \partial : N :: \mathcal{P}\rangle$ ***imports*** *x as P if either (i)* $\partial = \triangledown\flat$ *and* $[P : x] \in \mathcal{P}$ *or (ii)* $\partial = \lozenge\flat$ *and* $[P : \langle x, y\rangle] \in \mathcal{P}$ *for some y. We say that* $\langle x : \partial : N :: \mathcal{P}\rangle$ ***exports*** *x as P if either (i)* $\partial = \vartriangle\bigcirc$ *and* $[P : x] \in \mathcal{P}$ *or (ii)* $\partial = \lozenge\flat$ *and* $[P : \langle y, x\rangle] \in \mathcal{P}$ *for some y.*

We shall actually assume that what we previously called AISs are in fact PAISs, and the definitions of argument structure, merge and fusion will have to be lifted to the type of structure. Most of this actually goes without changing anything.

Recall that the referents of the left hand representation are indexed by 1, and the referents of the right hand representation are indexed by 2. In the phase of merge, unification of certain referents take place. Unification happens if two AISs are merged. Thus we shall look now at the substitution induced by a pair of merged AISs:

(4.29)
$$\begin{aligned} &\langle x : \Diamond\ominus : \nu :: \begin{bmatrix} \textsc{par}1{:}p_1 \mapsto p_4 \\ \textsc{par}2{:}p_2 \mapsto p_2 \\ \textsc{par}3{:}p_3 \mapsto p_5 \end{bmatrix} \rangle \bullet_r \langle y : \vartriangle\bigcirc : \nu :: \begin{bmatrix} \textsc{par}1{:}q_1 \\ \textsc{par}2{:}q_2 \\ \textsc{par}4{:}q_3 \end{bmatrix} \rangle \\ = \;& \langle x^1 : \vartriangle\bigcirc : \nu :: \begin{bmatrix} \textsc{par}1{:}p^1_4 \\ \textsc{par}2{:}p^1_2 \\ \textsc{par}3{:}q^2_3 \\ \textsc{par}4{:}p^1_5 \end{bmatrix} \rangle \end{aligned}$$

Additionally, the substitution $q^2_1 \mapsto p^1_1, q^2_2 \mapsto p^1_2$ must be applied to the second representation. This is because the AIS on the right exports q^2_1 under the name PAR1, and the left hand AIS imports p^1_1 under that name. And the right hand AIS exports q^2_2 under the name PAR2, while the left hand AIS imports p^1_2 under that name. The rules of merge are that in these cases the respective referents are to be considered the same.

The rule is this: suppose that the rightward merge succeeds. Then for every parameter name P the parameter that is exported by the righthard AIS under the name P is identified with the parameter that is imported bu the leftmost AIS under the same name.

We have assumed that those parameters that are not mentioned are simply passed on unchanged. So, the lexical entry for `big` can be expanded as follows.

(4.30)

/groß/
$\langle x : \Diamond\ominus : [\textsc{pers} : 3] :: \begin{bmatrix} \text{PROP} & : & p \mapsto p' \\ \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3 \end{bmatrix} \rangle$
$\varnothing$
$p' \doteq \mathsf{big}'(t_3)(p)$

However, the additional parameters may be suppressed as they are not needed.

We have seen earlier that certain adjectives determine whether or not the NP is formed at reference time or at story time. Their semantics therefore involves more parameters. (We allow the use of t in place of the longer statement $t \mapsto t$.)

/heutig/

(4.31)

$$\begin{array}{|c|}\hline \langle x : \Diamond\ominus : [\textsc{pers} : 3] :: \left[\begin{array}{lll} \text{PROP} & : & p \\ \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3' \mapsto t_3 \end{array}\right] \rangle \\ \hline \varnothing \\ \hline t_3' \doteq t_1 \\ \hline \end{array}$$

/damalig/

(4.32)

$$\begin{array}{|c|}\hline \langle x : \Diamond\ominus : [\textsc{pers} : 3] :: \left[\begin{array}{lll} \text{PROP} & : & p \\ \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3' \mapsto t_3 \end{array}\right] \rangle \\ \hline \varnothing \\ \hline t_3' \doteq t_2 \\ \hline \end{array}$$

Notice that none of these adjectives contributes to the property in question. They merely reset the predication time for the property. In both cases, there is an unused parameter; in the first case it is the story time parameter and in the second case the reference time. By our conventions on parameters these can be dropped. We shall remark here that the syntactic behaviour of these adjectives is not totally accounted for by its argument structure. Namely, these adjectives appear typically right after the determiner or the numeral.

(4.33) `der damalige erste Vorsitzende`

(4.34) `?der erste damalige Vorsitzende`

the of.that.time first chairman

the first chairman of that time

(4.35) `die vier damaligen stimmberechtigten Vereinsmitglieder`

`die damaligen vier stimmberechtigten Vereinsmitglieder`

`?die vier stimmberechtigten damaligen Vereinsmitglieder`

the four of.that.time with.right.to.vote club members

the club members of that time who had a right to vote

The same can be with the English word `former` or `alleged`. One explanation is that for the property that forms the NP it is required that it be homogeneous.

Hence, it is disfavoured to shift the time of predication in the middle of the NP. A different case are however the words like `ehemalig` ('former') or `Ex-` ('ex-'), which explicitly reset the predication time.

We end this discussion with an example from Nootka which is reported in Sapir [84] (our discussion is based on the passage in [18]). In Nootka, nouns may optionally be specified for whether they possess the indicated the property right now or whether they possessed that property in the past. The example given is

(4.36) `inikw-ihl-'minih-'is-it-'i`
fire-in:house-plural-diminutive-past-nominal
the former small fires in the house

4.3 Time and Tense

Let us take a closer look at time parameters. Notice that in a sentence, both the NP arguments of the verb may predicate a property of the actants that is different from the event time; moreover, it is conceivable that the predication of the properties are different for the individual arguments, as we have seen in the previous section, see the examples (4.21) – (4.26). As concerns the parameters of the actants, things are quite straightforward. The verb imports the referents for the actants but the relevant parameters are generally left unchanged. This means that the imported referent takes the triple of referent, story and event time directly from the verb. We will analyze below in detail how it changes them. Now, pending certain questions of detail to be answered, the entry for `run` is as follows.

/`run`/

(4.37)

$$\left\langle e : \triangle \bigcirc : [\textsc{inf} : \bigstar] :: \begin{bmatrix} \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3 \end{bmatrix} \right\rangle$$

$$\left\langle x : \triangledown \ogreaterthan : [\textsc{case} : \mathit{nom}] :: \begin{bmatrix} \text{PROP} & : & p \\ \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3 \end{bmatrix} \right\rangle$$

$$t_3$$

$$\mathsf{run}'(e); t_3 \in \mathsf{time}'(e); \mathsf{act}'(e) \doteq x.$$

We have written $t_3 \in \mathsf{time}'(e)$ to say that e happens at t_3. But if t_3 is actually an interval (which would be more realistic) then we should write $t_3 \doteq \mathsf{time}'(e)$. We shall see in a later section how the verb uses the property of the nominal argument. We shall henceforth employ the convention that parameters are shared in an item across all referents, unless specified otherwise. If this convention is employed, in the lexical entry for `run` no parameters need to be specified.

Now let us return to the question of tenses. We have already laid out the basic meaning of the three basic tenses: the present, the future and the past. We shall now consider this proposal in detail and extend it to other tenses. First, let us look at the present. Here is a particular entry for the present.

/PRES/

(4.38)

$$\left\langle e : \blacklozenge \olessthan : \begin{bmatrix} \text{CAT} & : \mathit{ev} \\ \text{TENSE} & : \bigstar \mapsto \mathit{pres} \end{bmatrix} :: \begin{bmatrix} \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \end{bmatrix} \right\rangle$$

$$\varnothing$$

$$t_1 \circ t_2$$

Here $t_1 \circ t_2$ means that t_1 and t_2 are cotemporaneous (meaning that they overlap). This means that $t_1 \cap t_2 \neq \varnothing$ in case they are intervals and $t_1 \doteq t_2$ in case they are time points. The entries for the past and the future are similar. The only difference is that they set the tense value to *past* and *fut*, respectively. The semantics is $\mathsf{before}'(t_2, t_1)$ and $\mathsf{after}'(t_2, t_1)$. The interpretation is roughly that the interval t_2

ends before t_1 begins—in the case of the past—and that t_2 begins after t_1 has ended. This particular translation can be justified for English (as opposed to those given by Comrie [18]) but this is rather irrelevant in the present circumstance.

There are also languages in which there exist more distinctions than simply between past, present and future. The following are the tense suffixes of Yandruwandra (see [18]):

(4.39)	`na`	very recent past
	`ṉana`	within the last couple of days
	`ṉukarra`	within the last few days
	`nga`	weeks or months ago
	`lapurra`	distant past

Here are the tense suffixes of Yagua (see [18]):

(4.40)	`jasiy`	proximate-1 (within a few hours)
	`jái`	proximate-2 (one day ago)
	`siy`	within a few weeks
	`tíy`	within a few months
	`jadá`	distant or legendary past

What these tenses add in addition to placing one interval with respect to another they also specify the distance between these intervals. It is clear that this can be encoded into the present framework. There are languages which only have these tenses, for example Hungarian, which has only present and past and an analytic future.

Now we turn to the complex tenses. In English, Finnish, German, Latin, and Greek, to name just a few, there exists a second series of complex tenses, namely the perfect, the pluperfect and the future perfect. Ignoring certain details one can say that they differ from the earlier ones in that the event time is before the story time, while in the simple tenses the event time is contemporaneous with the event time. In order not to get confused, we shall say that in addition to the tense feature previously introduced there is a secondary tense feature called TENSE2, which can have the values *pres*, *fut* and *past*, though it need not exist in all language nor does it have to have exactly these values.

$$/\textsc{pres}2/$$

$$(4.41)\quad \langle e : \blacklozenge\oslash : \begin{bmatrix} \textsc{cat} & :ev \\ \textsc{tense}1 & :* \\ \textsc{tense}2 & :\star \mapsto pres \end{bmatrix} :: \begin{bmatrix} \textsc{sto} & : & t_1 \\ \textsc{pred} & : & t_2 \end{bmatrix} \rangle \;\Big|\; \varnothing \;\Big|\; t_1 \circ t_2$$

$$/\textsc{past}2/$$

$$(4.42)\quad \langle e : \blacklozenge\oslash : \begin{bmatrix} \textsc{cat} & :ev \\ \textsc{tense}1 & :* \\ \textsc{tense}2 & :\star \mapsto past \end{bmatrix} :: \begin{bmatrix} \textsc{sto} & : & t_1 \\ \textsc{pred} & : & t_2 \end{bmatrix} \rangle \;\Big|\; \varnothing \;\Big|\; t_1 > t_2$$

Although in the languages mentioned above there is no equivalent for the future, there are languages in which such an element apparently exists. [18] reports that in Bamileke-Dschang "it is possible to have sequences of auxiliaries indicating time reference, though apparently two is the maximum number permitted in sequence. In such a sequence, the first auxiliary establishes time reference relative to the present moment, while the second auxiliary locates the situation relative to the reference point established by the meaning of the first auxiliary." Notably, Bamileke-Dschang has auxiliaries for the future tense and it distinguishes several future tenses that can be built up using future auxiliaries. Now, it is certainly possible that these tenses are simple tenses in this language, but it is equally likely that they are complex tenses established by stacking two tense features. (From the subsequent text it appears that the two markers function just like primary tense and secondary tense above.) It seems to us to be no coincidence that the limit in the sequence is two. There is to our knowledge no tertiary tense. If there were, we would have to introduce a fourth parameter to keep track of the additional intermediate time interval. However, in [18] Comrie wishes to put no upper bound on the number of stacked tense markers. He says that there is for example a future perfect in the past. Moreover, in French there exist so-called formes surcomposées, for example `avait eu fait` lit. *having had done*. However, we are reluctant to make an exception for them. For one, these forms are rare, and second even here there is no recursion. There is no *`avait eu eu fait`. Although this issue needs elaboration, we shall leave the matter at that.

It would now perhaps be wise to distinguish these various tenses. We shall say that there is a primary tense and a secondary tense. Each of the two can have various forms, present, past, future and possibly more. Some language have less tenses, some more. Finnish has a secondary tense but distinguishes only present and past. Hungarian has no secondary tense. So we have the following general scheme (taking tenses as suffixes, which is of course not necessary).

$$\begin{array}{ccccc} & & \vdots & & \vdots \\ & & \text{future2} & & \text{future1} \\ \text{verb} & + & \text{present2} & + & \text{present1} \\ & & \text{past2} & & \text{past1} \\ & & \vdots & & \vdots \end{array}$$

The perfect, pluperfect and future perfect of Latin are simple tenses, though they are formed using a different stem, the so-called perfect active stem. The analogous tenses of German and English are formed using the perfect active participle and an auxiliary. In languages which use the latter strategy, this auxiliary may be either `to be` or `to have`. We shall not be concerned with the selection of this auxiliary. What all these languages have in common is that the tenses of the second series are formed by different means than the corresponding simple tenses. Here is an example from Latin. The verb `tangere` (‘to touch’) has the present active stem `tang` and a perfect active stem `tetig`. The forms are active, 1st person singular indicative.

	`tang-ō`	`tetig-ī`
	touch.PRES2-PRES	touch.PAST2-PRES
	I touch	*I have touched*
	`tang-ēbam`	`tetig-eram`
(4.43)	touch.PRES2-PAST	touch.PAST2-PAST
	I touched	*I had touched*
	`tang-am`	`tetig-erō`
	touch.PRES2-FUT	touch.PAST2-FUT
	I will touch	*I will have touched*

This can be accounted for in the following way. We make the markers of present, past or future sensitive to whether the word they apply to has [TENSE2 : *pres*] or whether it has [TENSE2 : *past*]. So, the perfect stem itself already encodes the notion of the event being completed (ie that the event time precedes story time),

Table 4.1: The Complex Tenses

	pres2	past2	fut2
pres1	$r = s = p$ present	$p < s = r$ perfect	$s = r < p$?
past1	$p = s < r$ past	$p < s < r$ pluperfect	$s < r; s < p$ future in the past
fut1	$r < s = p$ future	$r < s; p < s$ future II	$r < s < p$?

while the present stem signal contemporaneity. The tense suffix has two forms, depending on whether it attaches to the present stem or the perfect stem, and we may therefore say that the tense suffix agrees with the stem in the secondary tense value. In the same way we can set up the tense systems of German, English and Finnish, which all use an auxiliary. We shall say that the auxiliary carries the primary tense and it applies only to a carrier of secondary perfect.

There are in total nine combination of these tenses. We have displayed them in Table 4.1. In this table we use r, s and p to denote the reference time, story time and predication time, respectively. Typically, in the Indo-European tenses, six out of these nine occur. However, English for example has a so-called future in the past, and other languages may also have additional tenses which are often not listed in the grammars.

As a special case of agreement we note an example reported in Comrie [18] going back to [81]. In Malagassy, certain adverbs must agree with the main verb. The word for *here* is `ao` in the present but `tao` in the past.

(4.44) `n-ianatra t-ao/*ao i Paoly omaly.`
PAST-study PAST-here DEF Paul yesterday

Obviously, in these adverbs there is a sensitivity for the tense. This has however nothing to do with the actual parameters, but simply agreement in tense.

We will end this chapter with a discussion of the notions of absolute and relative tense of Comrie [18]. Comrie notes that there is a distinction between a tense that is anchored in what he calls the deictic center and a tense that is anchored in a time point that is given in the text. He calls the first absolute and the second relative tense. We have called the deictic center the reference time. The distinction between absolute and relative tenses is captured here by the introduction of additional time points to which one can make reference to. For example, in Latin there are three participles, one denoting cotemporaneous action, one anterior action and the third future action. We can capture these facts by giving them the following semantics. (Recall that morphologically the participle is an adjective in Latin, and this is how it is represented here.)

(4.45)

`/tactus/`

$$\left\langle x : \Diamond\oplus : \begin{bmatrix} \text{CLASS} & : & \mathit{masc} \\ \text{CASE} & : & \mathit{nom} \\ \text{NUM} & : & \mathit{sing} \end{bmatrix} :: \begin{bmatrix} \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3' \mapsto t_3 \end{bmatrix} \right\rangle$$

e
$\mathsf{touch}'(e'); \mathsf{thm}'(e') \doteq x; t_3' < t_3; \mathsf{time}'(e') \doteq t_3'$

This say that there has been an event e' of touching x which happened at some time t_3' before the event e. This allows the participle to be an attribute, appearing either prenominally or postnominally. We shall return to this in the section below.

4.4 Reconsidering the Structure of the Noun Phrase

As we have changed the structure of the referent systems we shall investigate once again the structure of the noun phrase. Several issues need to be reconsidered. We shall assume that the complex consisting of adjectives and the head noun only specifies a property. Given this property, an individual or a group is being formed. This is done for example by using a numeral or other element designating quantity. Let us take an earlier example again.

(4.46) `quattuor magni mures`
four big-MASC.PL.NOM mouse-PL.NOM
four big mice

We shall ignore case for the moment. The lexical entries for `mus` ('mouse') is as follows:

/`mus`/

$$(4.47)\quad \begin{array}{|c|c|c|}\hline \langle x : \triangle\ \varnothing : & \begin{array}{lcl}\textsc{class} & : & \mathit{masc} \\ \textsc{num} & : & \star\end{array} & :: [\textsc{prop} : p]\rangle \\ \hline \multicolumn{3}{|c|}{\varnothing} \\ \hline \multicolumn{3}{|c|}{p \doteq \mathsf{mouse}'} \\ \hline \end{array}$$

Now the lexical entry for plural is like this

/PL/

$$(4.48)\quad \begin{array}{|c|}\hline \langle x : \blacklozenge\olessthan : [\textsc{num} : \star \mapsto pl] :: [\textsc{prop} : p]\rangle. \\ \hline \varnothing \\ \hline \\ \hline \end{array}$$

Notice that the plural suffix does not change the property parameter; nor does it contain any meaning. The lexical entry for `magn` ('big') is like this

/`magn`/

$$(4.49)\quad \begin{array}{|c|c|c|}\hline \langle x : \lozenge\obar : & \begin{array}{lcl}\textsc{class} & : & \star \\ \textsc{num} & : & \star\end{array} & :: [\textsc{prop} : p \mapsto p']\rangle \\ \hline \multicolumn{3}{|c|}{p' \doteq \mathsf{big}'(p)} \\ \hline \end{array}$$

The gender agreement morpheme has a straightforward semantics. Finally, we come to the numeral.

/`quattuor`/

$$(4.50)\quad \begin{array}{|c|}\hline \langle x : \lozenge\ogreaterthan : [\textsc{num} : pl] :: [\textsc{prop} : p \mapsto p']\rangle \\ \hline p' \doteq \lambda x.((\forall y)(y \in x \rightarrow p(y)) \wedge \sharp x \doteq 4) \\ p'(x) \\ \hline \end{array}$$

This semantics for the numeral works as follows. First, the property is lifted to a property not of individuals but of groups. Next a group is created, whose size is four and has the property of consisting entirely of *p*-ers.

It is important to note that it is the numeral that forms the group and which lifts the individual property to a group property. To attribute the group forming property to the plural would make the semantics unduly complicated. For a non-intersective adjective in the plural will expect from its head noun a group property and not an individual property. For example, the adjective `big` is a function from properties to properties. In the singular its semantics is

(4.51) $p' \doteq \mathsf{big}'(p)$

The semantics of `big` in the plural would then be as follows:

(4.52) $p' \doteq \lambda x.(\forall y \in x)\mathsf{big}'(p)(y)$

So, p' is the property of consisting entirely of big p-ers. Leaving the semantics unchanged would give the following result. The property `magni mures` would not be the property of being a set of big mice but the property of being a big set of mice. This is clearly not as it should be. Hence, the semantics of the adjective would have to be changed rather substantially when put into the plural. However, if we take plural not to form the group, matters are in fact quite straightforward.

Immediately, one problem appears. If things are construed as above there is actually no way to tell from the argument structure whether or not the group has actually been formed. We therefore need some device that tells us about that. There are several ways to achieve this, and it seems plausible that languages employ both of them. We might assume that it is possible *not* to pass on a parameter. This will happen exactly when the group is formed, because then it is not necessary to know about it any more. The semantical structure for a numeral is therefore

(4.53)

/`quattuor`/
$\langle x : \Diamond\ominus : [\textsc{num} : pl] :: [\textsc{prop} : p \mapsto *\rangle.$
$\varnothing$
$(\forall y)(y \in x \to p(y) \wedge \sharp x \doteq 4).$

A group is distinct from a property in that it bears no property parameter.

Another alternative is to relegate this matter to the presence or absence of the determinateness feature. A noun phrase is complete only when this feature is set, and it in turn can only be set after the group is formed—if a group is formed at all. Indeed, the present framework allows for several alternatives. The first is to assume that the determiner does nothing but to mark off the left edge of the phrase.

(4.54)

/`a(n)`/
$\langle x : \Diamond\ominus : \begin{bmatrix} \textsc{num} & : & sg \\ \textsc{def} & : & \star \mapsto - \end{bmatrix} :: p\rangle$
$\varnothing$
$\varnothing$

A different representation of `a(n)` is one that creates an object from the propertye. We may either assume that it thereby eliminates the parameter or that it does not. Here we assume that the property is gone:

(4.55)
$$\begin{array}{c}
\texttt{/a(n)/} \\
\hline
\multicolumn{1}{|c|}{\langle x : \Diamond\circleddash : \left[\begin{array}{lcl} \text{NUM} & : & sg \\ \text{DEF} & : & \star \mapsto - \end{array}\right] :: p \mapsto \star \rangle} \\
\hline
\multicolumn{1}{|c|}{x} \\
\hline
\multicolumn{1}{|c|}{p(x)} \\
\hline
\end{array}$$

Here the indefinite does nothing but to factually attribute the parametric property of the object! Notice that the variable x was doing no service at all during the composition of the noun phrase except in its function as a coherence device. We shall assume that it only has the latter function, namely attributing the property of the individual.

Next we look at the definite determiner. It may be used to convey the uniqueness of the object or its salience. In the first case its structure is:

(4.56)
$$\begin{array}{c}
\texttt{/the/} \\
\hline
\multicolumn{1}{|c|}{\langle x : \Diamond\circleddash : \left[\begin{array}{lcl} \text{NUM} & : & sg \\ \text{DEF} & : & \star \mapsto + \end{array}\right] :: p \rangle} \\
\hline
\multicolumn{1}{|c|}{x} \\
\hline
\multicolumn{1}{|c|}{p(x); (\forall y)(p(y) \rightarrow y \doteq x)} \\
\hline
\end{array}$$

In the plural, the marker for the indefiniteness is empty in English. Its semantics is the same as in the singular. It attributes the property to the group and asserts that the group is unique with this property. So, the phrase `the four mice` will be interpreted as a group consisting of four mice and which is unique in consisting of four mice. Notice that the determiners do not change the property parameter.

There is a list of quantifiers that provides additional evidence for the existence of properties. These are the so-called proportional quantifiers like `few`, `many`, `most`, `three quarter of`, `all`. What is common to them is that they do not specify an absolute quantity but a quantity that is relative to the size of the largest group.[1] Take for example `all`. A group consisting of all soldiers is a group

[1] We note here that `few` also has an absolute reading. For example, `a few soldiers` means a small group of soldiers, while `few soldiers` usually means a small group of soldiers compared the number of soldiers.

comprising all individuals that are soldiers. Without knowing who is and who is not a soldier it is impossible to form that group. Alternatively, and this is the line we are taking here, the group consisting of all soldiers is the set formed by using the property of soldierhood:

(4.57)
$$\begin{array}{c} /\texttt{all}/ \\ \boxed{\langle x : \Diamond\ominus :} \boxed{\begin{array}{lll} \text{NUM} & : & pl \\ \text{DEF} & : & \star \mapsto - \end{array}} \boxed{:: p \mapsto p' \rangle} \\ \boxed{\begin{array}{c} x \doteq \{y : p(y)\} \\ p' \doteq \lambda x.(\forall y \in x)p(y) \end{array}} \end{array}$$

So, `all` forms the group of all things satisfying the property. Notice that it also sets the definiteness value to −. Even though it forms a group, this group is not definite. With this, the ungrammaticality of the following example is accounted for:

(4.58) *`Watson read these/the all newspapers.`

This is so since the determiner needs as a complement a phrase with undefined definiteness value. Yet, the definiteness is already set, so no determiner may be present. Notice that there is a construction, shown in (4.24), which involves `all` and is nevertheless grammatical.

(4.59) `Watson read all (of) the newspapers.`

(4.60) `Watson read few/most/many of the newspapers.`

Similarly with the numerals. This use is most easily accounted for by allowing them to take a full definite plural NP as a complement. This NP must be in the genitive. The expression `three quarter` allows only the latter type of construction and may not be used with a property:

(4.61) *`Wayne sent three quarter soldiers to the camp.`

This shows that although the two constructions—taking a property as a complement or a definite plural NP—are related, they are syntactically independent and elements may individually choose to occur in just one or both of the construction types.

With the definiteness value set, the noun phrase may or may not be complete. If the NP is indefinite, then it is already complete. If the NP is definite, it may additionally receive what we call for want of a better name a **proximity value**. In English, a definite NP can be formed using either the plain definite determiner `the` or the words `this` or `that`. We shall assume that they set the proximity value to ± (there may be more values in other languages). These words may also considered as deictic words. In Hungarian the deictic element is not part of the case domain as can be seen from the following examples.

(4.62) `Voltam a/egy házban.`
be-PAST-1.SG DEF/a house-INESS
I was in the/a house.

(4.63) `Voltam ebben/abban a házban.`
be-PAST-1.SG PROX-INESS/REM-INESS DEF house-INESS
I was in this/that house.

(4.64) `Minden házban volt egy cica.`
Every house-INESS be-PAST-3.SG a cat
A cat was in every house.

(We gloss `ez` as PROX (proximate) and `az` as REM (remote). They do not set the definiteness value. This is done by `a(z)`. The English (near) equivalent `this` would then have to be glossed as PROX.DEF and `that` would be REM.DEF.) Case must be repeated after the deictic element.

We will now address a topic that has been kept in the background for most of time, namely the relationship between morphological and syntactical bracketing. The default assumption, namely that morphological bracketing is just part of the syntactic bracketing, can be shown to be problematic for many reasons. One is a semantic one. Take the adjective `former` and the prefix `ex-`. Both have the same semantics, but one is a separate word while the other is only part of a word. If the syntactic bracketing and the morphological bracketing coincide we would not expect the following two to mean the same.

(4.65) `Peter is the former director of the Bank of Scotland.`

(4.66) `Peter is the ex-director of the Bank of Scotland.`

However, both mean the same thing and therefore `ex-` takes scope over the phrase `director of the Bank of Scotland`. The semantics that we have developed is however in large parts associative and therefore there is in this case no need to assume that the syntactic analysis is distinct form the morphological one. Nevertheless, there are cases when the semantics is not associative. One such case is the composition of the Hungarian noun phrase. Here, case and plural marking are suffixed to the head noun, which is at the end of the NP. Therefore, the adjectives, quantifiers, numerals and the determiner do not show agreement at all. We have previously argued that this is a morphological fact. In the morphology it is specified that only nouns inflect for number and case. (This applies however also to the deictic words `ez` (this) and `az` (that).) Now we are in a conflict. An inflected noun needs the adjective to agree with it in the features in which it inflects. But there is no overt agreement. We could argue at this point that adjectives do inflect for all these categories but all forms are identical. If this is assumed we have no problem, we can simply proceed as if Hungarian was like German or Finnish. However, it does not seem to us not the most obvious of all solutions. It also is historically incorrect. It is known that many cases, for example the inessive, have once been inflected nouns. If only morphological cases are iterated, we must assume that at that stage there was no agreement for the inessive. All that happened after that was that the postposition got weaker and eventually became a suffix. It is quite absurd to assume that Hungarian has implemented full case agreement when no such stage can ever be attested.

Thus, we assume that in Hungarian adjectives and determiners do not inflect. Since there is no direct evidence to distinguish these two approaches we shall argue from a historical point of view. If we assume that the categories in which a language categorizes elements from a morphological point of view are by and large arbitrary then we must assume that those categories that the morphology does not use at all are simply undefined rather than being defined but underdetermined. Suppose however that a category exists in the form of a distinct element, for example a postposition, that gradually reduces to, say, a case ending. From the standpoint of the system we previously had no reason to suspect that words are discriminated for case (take by way of example a language like English, Chinese or Tagalog). Once the morphology has changed and the postposition has been reduced to a case, we do however have a new morphological category, namely case. Now, what shall we say: is case a category of all words or just of some, for example the head noun? I think there is every reason to believe the second. The first option would be the result of a development when for example case distinctions are gradually lost (as in English) and the system may still list them as distinct cases, while their forms are already nondistinct. (The English nominative and accusative is a case in point. The two cases are only distinct in the pronouns.) This state of affairs is highly instable, as one might suspect, and will be reshaped into one where the irrelevant distinctions are eliminated. Moreover, once a category has lost all distinctions it may simply be removed.

We conclude from this discussion that it may well be that case morphology is selective in certain categories and that case may be undefined in others. Applied to Hungarian this means that case and number are undefined for the adjective, the numeral, the determiner and the quantifiers. But if that is so, the adjective can no longer combine with an inflected noun. Its case value is ⋆, but that of the complement noun is defined. The solution to this problem is to assume the following analysis:

(4.67) `(ez)-ek-ben (a fehér ház)-ak-ban`
(this)-PL-INESS (DEF white house)-PL-INESS
in this white house

Let's assume that the Hungarian plural and case suffixes are not word affixes but phrasal affixes. How can this be achieved? A simple mechanism is to assume that (nominal) case and number markers select a complement that has a defined definiteness value (which may be either definite or indefinite) but whose proximity value is undefined. The consequence is that the noun phrase must be finished up to the determiner `a(z)` before the case ending is attached. Moreover, the case ending must be attached there. That the proximity marker also carries case can be explained by the fact that case agreement is mandatory if it wants to combine with the NP, because that NP has the case and number features instantiated. However, we must obviously assume that it actually can inflect for these categories and therefore we must assume that case attaches also to elements in which proximity and definiteness are defined. Moreover, there are nouns which inflect for case in particular the demonstratives `ez` and `az`. If that is so, the following is expected to be grammatical as well.

(4.68) *`(ez a fehér ház)-ak-ban.`
(this DEF white house)-PL-INESS
(lit.) in this white houses

To solve this problem, we shall assume that we have two kinds of affixes, one being a word affix and the other being phrasal. The final nominal case and number suffixes are phrasal (as are the possessive markers), while the case and number

Figure 4.1: Phrasal and Word Case in Hungarian

/bVn/	/bVn/
$\left\langle x : \lozenge\olessthan : \begin{bmatrix} \text{CASE} & : & \star \mapsto \mathit{iness} \\ \text{DEF} & : & \top \\ \text{PROX} & : & \star \end{bmatrix} \right\rangle$	$\left\langle x : \blacklozenge\olessthan : \begin{bmatrix} \text{CASE} & : & \star \mapsto \mathit{iness} \\ \text{DEF} & : & \top \\ \text{PROX} & : & \top \end{bmatrix} \right\rangle$
$\varnothing$	$\varnothing$
$\varnothing$	$\varnothing$

markers that are suffixed to the demonstratives and the proximity markers are actually word affixes. They are distinguished as follows. The phrasal suffix needs the proximity value to be $\star$. The word affix on the other hand requires the proximity value to be different from $\star$. The inessive case suffixes are shown in Figure 4.1. Notice that the word affix is fusional, the phrasal affix nonfusional. By this assumption, the example (4.33) is ruled out because the phrasal case affix needs an undefined proximity value. Notice that the same problem appears in the English NP. Here, as there is not much of a case distinction left, there is nevertheless the category of number. However, number is marked at the NP only at the head noun, and in addition at the proximity markers (`this`/`these` and `that`/`those`). The indefinite article also two forms (`a(n)`/∅). In English we must assume that number is a phrasal affix which attaches to a phrase that has its definiteness value undefined. We may however assume that numerals take complements with a number value assigned to them. Therefore the bracketing of the English NP is as follows.

(4.69) `these four unsolved thorny problems`
this-PL four (unsolved thorny problem)-PL

This is the only bracketing possible, since otherwise the the adjectives cannot combine with their complements.

Notes on this section. As we have argued earlier (see 2.9) the semantics of the actual inflection marker is empty. However, there are exceptions to this rule. In Hungarian the plural marker is obligatorily absent in the presence of a numeral. Thus, the plural marker signals a multitude, just as the numeral `negy` (‘four’) signals ‘four’. Let me also briefly remark on the issue of pluralia tanta. The difference beween pluralia tanta and ordinary nouns is that the former are listed in the lexicon without a root form. For example, the Latin word `litterae` is ambingous between the plural of `littera` ('letter (of the alphabet)') and the pluralium tantum `litterae` ('letter'). The lexicon contains both `littera` ('letter of the alphabet') as a root noun and `litterae` ('letter') which has the argument structure of a plural noun.

4.5 Predicative and Attributive Adjectives

Adjectives occur basically in three types of environments. They can be modifiers of a noun, they can modify verbs (in which case they are called *adverbs*; we shall group both categories together here). They can be used predicatively, for example in postmodifiers in English or in postcopular position, and finally they can occur in what is syntactically often analysed as a small clause. Each of these constructions is distinct, and one can find that languages group them in different ways, as we have seen earlier. Here we shall be concerned with the implications of these facts for the semantic structure of adjectives. Let us first illustrate these types of contexts.

(4.70) `John is a clever student.`

(4.71) `John is running fast.`

(4.72) `John, proud of his achievement, went into the office.`

(4.73) `John is clever.`

(4.74) `John drove the car drunk.`

(4.75) `John drank himself stupid.`

(4.70) is once again an adjective modifying a noun, (4.71) an adverbial. (4.72) shows an adjectival phrase in postnominal position. Typically, this construction is used to make another assertion, one whose connection with the main assertion can only be guessed (here it seems simply that the two are contemporaneous). (4.73) is a case of a postcopular adjective and (4.74) is a depictive. In the syntactic literature this construction is analysed as a small clause type of construction, (4.75) is a resultative.

Certain things need to be noted. First, none of these constructions is restricted to adjectives (PPs or NPs can also serve in them), and second adjectives cannot always be put into all four contexts. A good example is `alleged`, which refuses to appear in postcopular position or as an adverbial. So, some care has to be exercised with respect to the generalizations that will arise from the semantics. The basic problem with adjectives is that their representations only licenses them to appear as nominal modifiers. They also cannot be in postnominal position because they are prenominal modifiers in English. They cannot be in postcopular position because they need a noun to modify and there is none. They can also not be depictives or resultatives. That this is no accident is corroborated by the fact that these constructions are marked by morphological distinctions. In German, the adjective inflects only when used as a prenominal modifier. Otherwise, it takes one and the same form. We might therefore propose that the other three construction types require an adverbial. However, we consider an adverbial only a modifier of a verb, and by this criterion the postcopular and the postnominal attribute is certainly not an adverbial. In Hungarian, the adverb is distinct from the adjective and is used only is the true adverbial context. (Note that the copula is zero in the third person.)

(4.76) `János csendesen dolgozik.`
János silent-ly works.
Janos works quietly.

(4.77) `Ez a motor csendes.`
PROX DEF motor (is) silent.
This motor is quiet.

(4.78) `Ezek a motorok csendesek.`
PROX-PL DEF motor-PL silent-PL
These motors are quiet.

We see therefore that the constructions must be kept distinct. (In English this is generally also the case; however, certain verbs do not require the adverbial form, like `drive`, and some adjectives are nondistinct from their derived adverbs like `fast`.)

In Finnish the adjective must appear in the essive if it is used in a depictive and in the translative if used in a resultative.

(4.79) ???

Now the semantic difference is as follows. While the postnominal adjective does not take part in the formation of the group it functions practically as a separate assertion on the group. We may analyze postnominal adjectives as if they head separate clauses. One difficulty in accounting for the various facts surrounding the adjective is that in some languages they inflect in some positions and not in others, and in another languages it might be different. In Hungarian, the adjective does not inflect in prenominal position, but it does in postcopular position. In German it is the converse. In French it inflects in both positions. In Georgian it inflects differently when used postnominally (see Section 3.3). Another difficulty is that adjectives appearing in postcopular position function as if they are nouns. For if the copula takes the adjective as one argument and the subject as another, there is still the complement of the adjective missing in the construction. The construction would be incomplete in this way. We shall therefore assume that the adjective appears with a dummy property inserted, which may for example be equated with the property that the subject provides. For example, take the sentence

(4.80) `This mouse is big.`

We shall assume that either the mouse is said to be big in the absolute sense or that it is big in the sense of being a mouse (or in another contextually given sense). It is the latter interpretation that interests us. Take the lexical entry for the English adjective `big`.

(4.81)
$$\begin{array}{|c|}\multicolumn{1}{c}{/\texttt{big}/} \\ \hline \langle x : \lozenge ⧁ : \begin{bmatrix}\textsc{cat} : ob \\ \textsc{num} : \star\end{bmatrix} :: p \mapsto q\rangle \\ \hline \varnothing \\ \hline q \doteq \mathsf{big}'(p) \\ \hline\end{array}$$

Let us assume that there is an empty element ONE that acts as an argument to the adjective.

(4.82)
$$\begin{array}{|l|}\multicolumn{1}{c}{/\textsc{one}/} \\ \hline \langle x : \vartriangle \bigcirc : [\textsc{cat} : ob] :: p\rangle \\ \hline \varnothing \\ \hline \varnothing \\ \hline\end{array}$$

It follows that the adjective together with ONE has the following structure

(4.83)
$$\begin{array}{|l|}\multicolumn{1}{c}{/\texttt{big} + \textsc{one}/} \\ \hline \langle x : \vartriangle \bigcirc : [\quad] :: q\rangle \\ \hline \varnothing \\ \hline q \doteq \mathsf{big}'(p) \\ \hline\end{array}$$

Next, we shall assume that the copula has the following form

(4.84)

/be/
$\langle e : \vartriangle \bigcirc : \mathfrak{y} \rangle$ $\langle y : \triangledown \circleddash : \mathfrak{z} \rangle$ $\langle x : \triangledown \circleddash : \mathfrak{z} :: p \rangle$
e
$p(y)$

Several things must be noted. $\mathfrak{y}$ and $\mathfrak{z}$ are metavariables for AVSs. They are just proxy for AVSs. The name $\mathfrak{z}$ is shared across the copula from x to y. This accounts for agreement in all relevant categories between the subject and the postcopular adjective, as is the case in French. Lack of agreement must still be accounted for, however. The second thing to note is that the event variable plays no role here. It is there for formal reasons (to attach tense, for example). Yet, semantically it is needless. In fact, the above description is somewhat inaccurate and we should write $p(y)(t)$, adding t as an additional tense parameter. The third is that the structure of the construction is that the copula takes the postcopular adjective first and then the subject.

There is now an immediate problem in that the copula does also allow a full NP to appear in postcopular position. Even though one could dismiss this fact as a distinct use of the copula, the regularity with which this construction is found across languages as well as the fact that semantically these structures are similar, call for a revision in the analysis. From a semantic point of view, both constructions are the same: some property is attributed to someone or something. This property can be expressed by an adjectival phrase or by a noun phrase. If this is correct then we should expect those NPs that do not denote properties not to appear in postcopular position. This seems to be the case.

(4.85) `John is a fool.`

(4.86) `John is the biggest fool on earth.`

(4.87) `*John is every husband.`

(4.88) `They are the soldiers.`

(4.89) `?They are a few soldiers.`

(4.90) `?They are most of the soldiers.`

Quantified NPs are generally disallowed in postcopular positions. The reason, as we see it, is that they do not denote properties. [2] We hold it that constructions of the kind `X is Y` ascribe the property of being identical to `Y` to `X`.

(4.91) `John is the dean of this faculty.`

(4.92) `Tully is Cicero.`

On the other hand, the subject must denote an individual or quantify over individuals. The requirement that the subject be an individual is not so strict, however.

Now, in order to be able to prevent the quantified NPs from appearing and in order to assign the proper semantics to those NPs that do we must in fact assume that NPs come in two kinds: as object denoting NPs and as property denoting NPs. We shall therefore, for want of a better solution, introduce a feature PROP with values + for a property and − for a non-property, ie an individual or a group. The idea is that only entities with [PROP : +] can appear in postcopular position. To make this work, we shall take it that adjectives are [PROP : +] together with simple nouns, and that the numeral or quantifier resets this value to [PROP : −]. The property feature is therefore an indicator of whether or not a group or an individual has been formed or whether the NP is taken to denote a property. The determiners can do both. The indefinite can be used to form an NP denoting an individual, while it can be used to form a property as well. Likewise the definite determiner, although there is a preference to use it to create individuals. But note the use in (4.86) of the definite determiner in connection with adjectives in the superlative. A different solution is to adopt a new empty element which can change an NP into a property:

(4.93)

<table>
<tr><td colspan="3">/PRP/</td></tr>
<tr><td>$\langle x : \Diamond\ominus :$</td><td>$\begin{bmatrix}\text{DEF} & : + \\ \text{PROP} & : - \mapsto +\end{bmatrix}$</td><td>$:: [\text{PROP} : p \mapsto q]\rangle$</td></tr>
<tr><td colspan="3">$\varnothing$</td></tr>
<tr><td colspan="3">$q \doteq (\lambda y.y \doteq x)$</td></tr>
</table>

Here, only the relevant details are shown. Notice the interplay between the parameters and the objects. The object x disappears in the semantics (even though formerly still present), while it is recoded as the property of being identical to x.

[2] I can see as an exception to this only the fact that a quantified NP can ascribe that the subject contains that many individuals of the described property. This is a plausible reading for these sentences. This reading would have to be accounted for, but our present discussion provides no means for doing so.

Notice that it is required that the NP is definite. This makes sure that the object x has been formed. Moreover, it would fail badly if it were applied to indefinite NPs as well.

The distinction between properties and individuals is also useful for a number of verbs that rather than taking an object as argument require a property. A clear example are `to call`.

(4.94) `The people call Arno a master.`

It is clear that `a master` is not an indefinite NP but rather a property attributed to Arno. An interesting fact about such verbs is that in certain languages the property denoting NP shares the case with the object.

(4.95) `Die Leute nennen Arno/ihn einen Meister.`
The people call Arno-ACC/him-ACC a-ACC master-ACC

(4.96) `Arno/Er wird von den Leuten ein Meister genannt.`
Arno-NOM/he-NOM is by the people a-NOM master-NOM called.

In our framework this can be achieved by letting the subject and the property share the same variable, which must therefore transport all its features. However, certain adaptation would be necessary to make this proposal viable.

Now we turn to the adverbs. What the adverbs have in common with the postcopular adjectives is that they are construed without a complement. Hence, we shall assume that they are construed with the help of the element ONE. In contrast to nominal modifiers the adverbs must also determine which of the arguments they want to modify. This is called **orientation**.

(4.97) `Walter is driving the car fast.`

(4.98) `Walter is driving the car drunk.`

In (4.97) the adverb `fast` modifies the speed of the car, not that of Walter (he could use telecontrol, for example). In (4.98) it is Walter who is drunk, not the car. In the first case we speak of **object orientation** and in the second case of **subject orientation**. Notice however that orientation is not determined by grammatical status. It changes with diathesis. If an adjective show object orientation in an active sentence it shows subject orientation in a passive. In German there are also exists impersonal passive. An adjective showing subject orientation in an active sentence can be used in the impersonal passive:

(4.99) `Johann warf den Ball weit weg.`
John threw the ball far away

(4.100) `Der Ball wurde weit weg geworfen.`
The ball was far away thrown

(4.101) `Gesine tanzte schön.`
Gesine danced nicely

(4.102) `Es wurde schön getanzt.`
it was nicely danced

These facts motivate the proposal that there is an element ADV that turns an adjective into an adverb. ADV must be preceded by ONE. Its semantics is in first approximation as follows.

(4.103)

/ADV/
⟨*e* : ◊♭ : ŋ⟩ ⟨*x* : ∇⧀ : ʒ :: *p*⟩
∅
$p(x)$

There are a number of amendments that need to be made. First, the adverbs can be of different kind and therefore contribute different meanings. For example, `fast` is a manner adverb, while `drunk` is not. We shall offer no solution to this problem. The second is that we must also assign x the right value. Obviously, we must have means to associate x with some actors of the event, either by means of θ-role (as we did previously), or by means of grammatical functions. Before this is done, it must be decided which is the argument with which the nominal is or can be construed. Again this is an issue which we shall not solve here. Obviously however, it is no problem to incorporate θ-role driven construal into the semantics as it is now. Consider for example an adverb like `hastily`. It seems that this adverb is invariably construed with the actor of a sentence rather than with the subject:

(4.104) `The informant hastily gave the spy the papers.`

(4.105) `The spy was hastily given the papers by the informant.`

For such an adverb the following version of ADV is appropriate:

(4.106)
$$\begin{array}{c} /\text{ADV}/ \\ \hline \langle e : \Diamond\mathfrak{b} : \mathfrak{y}\rangle \\ \langle x : \triangledown\varolessthan : \mathfrak{z} :: p\rangle \\ \hline \varnothing \\ \hline p(x); \mathsf{act}'(e) \doteq x. \end{array}$$

The element ADV is actually overt in many languages. In English it appears as `-ly`, in Hungarian it is `-an`/`-en` or `-ul`/`-ül`, depending on the adjective and the vowel harmony. In German, the adjective is turned into an adverb by using the bare, non inflected form. We shall assume, though, that it is already the element ONE that turns the adjective bare, and the ADV is actually zero. This explains why adjectives do not inflect in postcopular position.

Finally, we shall turn to resultatives. This is a very interesting construction. The resultative introduces a result state of the event; moreover, it adds a new transitive object to the verb. In German, the verb plus resultative behaves just like a transitive verb; the resultative object can be passivised and scrambled. Furthermore, resultatives that consist of a directional PP show the same behaviour for this PP.

(4.107) `Peter trank seinen Kumpel unter den Tisch.`
Peter drank his buddy under the table

(4.108) `Peters Kumpel wurde unter den Tisch getrunken.`
Peter's buddy was drunk under the table.

(4.109) `Seinen Kumpel trank Peter unter den Tisch.`
his buddy Peter drank under the table

(4.110) `Unter den Tisch trank Peter seinen Kumpel.`
under the table drank Peter his buddy

(4.111)
$$\begin{array}{c} /\mathsf{res}/ \\ \hline \langle x : \triangledown\varolessthan : \begin{bmatrix} \text{CAT} & : ob \\ \text{NUM} & : \star \\ \text{CASE} & : acc \end{bmatrix} :: p \mapsto q\rangle \\ \langle e : \blacklozenge\varolessthan : \begin{bmatrix} \text{CAT} & : ev \\ \text{VOICE} & : \star \end{bmatrix}\rangle \\ \hline \varnothing \\ \hline \mathsf{res}'(e, p(x)). \end{array}$$

This structure has interesting behvaiour. It combines with a verb that has not undergone diathesis. It combines via fusion and adds a transitive object. This means for German that the resultative must be added before the verb is actually inflected. This allows to incorporate the transitive object and to make it into a regular argument of the verb. It follows that it can undergo diathesis and all other regular processes of relation change.

4.6 Sequence of Tense

In recent years, there has been growing attention to the problem of what is known as **consecutio temporum** or **sequence of tense** (see [1] and [73]). The problem is simply put the following. In subordinate clauses, tenses do not necessarily take the reference time of the main clause as their reference time, but may instead choose to set the reference time differently. For example, Russian differs from English in that the subordinate clause sets its reference time to the event time of the main clause, while in English the reference time is not adjusted. The difference comes out clearly in the following example.

(4.112) `Pjetja skazal, čto Misha plačet.`
Pjetja said that Misha is crying
Pjetja said that Misha was crying.

This shift in tense does not appear in relative clauses:

(4.113) `Pjetja vstretil čeloveka, kotory plačet.`
Pjetja met a person who is crying
Pjetja met a person who is crying.

How do we account for the different behaviour of tenses in Russiand and English? Recall that verbs like *say*, `promise` and so on select a tensed subordinate clause. They may therefore adjust the parameters of the subordinate clauses. Therefore, the following appears in the argument structure of the verb `to say`.

(4.114) $$\langle e : \vartriangle\ \varnothing : \alpha :: \begin{bmatrix} \text{REF} & : & t_1 \\ \text{STO} & : & t_2 \\ \text{PRED} & : & t_3 \end{bmatrix} \rangle, \langle e' : \vartriangle\ \flat : \alpha' :: \begin{bmatrix} \text{REF} & : & u_1 \\ \text{STO} & : & u_2 \\ \text{PRED} & : & u_3 \end{bmatrix} \rangle$$

There are six parameters, three for the main clause and three for the subordinate clause. Since the tenses of the subordinate clause fix u_2 and u_3 with respect to u_1, we minimally need to give a value to u_1. The different choices are to set u_1 to one of t_1, t_2 and t_3. Suppose that u_1 is set to t_1. Then the reference time of the subordinate clause is the same as the reference time of the main clause. In this case we have to use past tense if the event of the subordinate clause happens at the same time as the one of the main clause and the main clause is in the past tense. This is the situation in English. If we set u_1 to t_2, then if both events happen at the same time, the subordinate clause is in the present tense. This is the situation in Russian. The same would happen if we took u_3 to be t_3. The results would be different if the main clause was in the pluperfect. We are not in a position to test the difference, however.

The situation is however somewhat more involved than that. Here is an example.

(4.115) `Yesterday, John decided that tomorrow morning he would start working.`

The embedded event happens in the future, seen from the perspective of the main clause. Yet, we do not get the future tense, but what is known as *future in the past*. This tense is put when the story time is in the past from the reference time but the predication time is in the future of the story time. We conclude therefore that in English the subordinate clause not only fixes the value of the reference time of the subordinate clause, but also the story time. The story time is set to the predication time of the min clause (the time of John's decision). The reason why we get the future in the past is not the following. The verb in the embedded clause must be tensed, and the tense must be such that the story time of the embedded clause is anterior to its reference time. However, the predication time is after the story time (and also after the reference time, but that does not count here), and so the resulting tense is future in the past.

It is expected that if the reference time of a sentence is reset, the time referred to by temporal adverbials is shifted as well. However, we find that there are three classes of adverbials. The first class may be called **event relative**, the second **utterance relative** and the third **absolute**. Absolute adverbials are dates, such as `1st of May`, `in 1900` and so on. Their interpretation is fixed, and does not depend on the utterance or and other time point in the sentence. The semantics of these expressions is the same in all languages. To give an example, [3] Let us assume that today is the 13th of May. On the 8th of May Kolya says

(4.116) `Ja pridu četyrnadcatogo maja.`
I will arrive on the 14th of May.

If this is reported today, one would have to say

(4.117) `Kolya skazal, čto on pridet četyrnadcatogo maja.`
Kolya said that he would arrive on the 14th of May.

Utterance relative adverbials are `now`, `tomorrow`, `yesterday`. When they are used, they fix time point relative to the point of utterance. For example, by using `yesterday` in the main clause and `tomorrow` in the subordinate clause in the example (4.61), it is guaranteed that the predication time of the main clause is the day before the utterance, while the predication time of the subordinate clause is the day after the utterance, in particular it happens after the predication time of the main clause (which is why the future in the past is obligatory). It the example we expect that we can exchange the expression `16th of May` with `tomorrow`. In English this is fine. The same in Russian:

(4.118) `Kolya skazal, čto on pridet zavtra.`
Kolya said that he would arrive tomorrow.

However, as Comrie notes, if today was the 15th of May and not the 13th, then we can say

(4.119) `Kolya skazal, čto on pridet četyrnadcatogo maja.`
Kolya said that he would arrive on the 14th of May.

but we cannot say

(4.120) `Kolya skazal, čto on pridet včera.`
Kolya said that he would arrive yesterday.

This, he explains, is a fact of Russian grammar. It is not allowed to collocate an adverbial with past reference with a future tense. What is crucial is that the past

[3] These examples are taken from Comrie [18].

reference must be overtly marked on the adverbial, and not simply accidental, as with absolute adverbials.

The third class of adverbials are the event relative adverbials. These are `the day after`, `the day before`, `on that day`. Hence we find the following.

(4.121) `Džon skazal: ‘Ja ujdu zavtra.’`
John said: ‘I will leave tomorrow.

(4.122) `Džon skazal, čto on ujdet na sledujuščij den.`
John said that he would leave the following day.

Here, although the tenses in the subordinate clauses are different, as explained above, the adverbials function in the same way. They fix the predication time relative to the story time. We have already said that the story time of the subordinate clause is set to the time of John's uttering that sentence, which is the predication time of the main clause. The adverbial `the following day` establishes that the predication time is one day after the story time.

The semantics of these adverbials is a tricky matter. Notice that if Russian allows the reference time of the subordinate clause to be shifted, there would be no possibility for the adverbials to pick up the reference of the main clause. Thus, we would predict that no utterence relative adverbials can exist in Russian. This is not what the above data shows. Instead, we assume the following. When an utterance is made, speaker and hearer are in a certain situation σ. This situation consists minimally of a speaker, a hearer and the utterance. It is located in space and time. Let us introduce functions spk', hr', time', and loc', which, given a situation σ, yield the speaker, the hearer, the time and the location of that situation. So, when an utterance u is made, there is this event, σ of uttering u. We will use σ as a unique variable to denote the situation. We shall also assume that there are constants such as $\mathsf{14\text{-}May\text{-}1999}'$, which yield the time interval of the 14th of May 1999. And finally we assume that there are relations such as $\mathsf{days\text{-}after}'(t_1, t_2, n)$, where t_1 and t_2 are time points (or intervals) and n a number. This holds if t_2 is n days after t_1. With these elements given, we shall give a semantics for the adverbials of English and Russian. The absolute adverbials show the same behaviour in Russian and English. Their denotation makes no reference at all to the internal time points.

(4.123) `/on 14th of May 1999/`

$\langle e, \Diamond\obar, \mathfrak{z}\rangle$
$\varnothing$
$t_3 \subseteq$ 14-May-1999′

(The directionality shall not be of importance here.) Notice that we could also have construed this adverbial as saying that the event time of e is contained in the said interval. Utterance relative adverbials likewise show no difference in behaviour in the two languages.

(4.124) `/yesterday/`

$\langle e, \Diamond\obar, \mathfrak{z}\rangle$
$\varnothing$
days-after′(time′(σ), t_3, 1)

The semantics says that the predication time is one day before the time of the utterance. (Actually, it says that the utterance time is one after the predication time, but that amounts to the same.)

Finally we go to the event relative adverbials. Here we find that no reference is made to the utterance time or some absolute time point. And this means that their meaning is shifted when the reference time is shifted as well.

(4.125) `/the following day/`

$\langle e, \Diamond\olessthan, \mathfrak{z}\rangle$
$\varnothing$
days-after′(t_3, t_1, 1)

Notice that the meaning of the adverbials does not differ in Russian and English. The sequence of tense is the only difference between the two languages.

In addition to these classes of adverbials there is also a class of adverbials that are simply anaphoric. This means, they can connect two events that are otherwise not syntactically related. An example is the following.

(4.126) `John left Paris. One day later he returned.`

The treatment of such adverbials is pretty much the same as that of pronouns. The adverbial picks up the reference time of the preceding sentence and resets it for the next sentence:

(4.127)

`/one day later/`

$\langle e, \Diamond\ominus, [\text{FORCE} : \checkmark] :: [\text{REF} : u_1 \mapsto t_1]\rangle$
$\varnothing$
days-after$'(t_1, u_1, 1)$

(Actually, this is how the semantic structure looks like for a clause initial temporal adverbial.)

4.7 Sequence of Persons, Worlds and other Indices

As Schlenker [85] observes, it is not only time points that obey certain rules of succession, but also worlds, persons and locations. It is our purpose to survey some of the possibilities here and not to lay down a comprehensive theory of them. We shall simply point out just what the phenomenon consists in and how it can be analyzed using parameters. It will be seen that the basic mechanics is the same, whether we take time points, worlds, persons or other. The first set of examples deal with the change of person in embedded speech. Suppose we have the following instance of direct speech:

(4.128) (John says:)`‘I am sick.’`

Then there are — in principle — two ways of turning this into reported speech:

(4.129) `John says that he is sick.`

(4.130) `John says that I am sick.`

English represents a language of the first kind. (4.129) is grammatical if taken to represent (4.128), but (4.130) is not. Instead, if I were to say that I am sick, then (4.130) must be used with `I` in place of `John` and (4.129) (changed similarly) is unacceptable.

There are languages, in which we must use (4.130). Such a language is Amharic. (4.131) is intended to show that what we are dealing with is not direct speech.

	`mən`	`ʿamtʿa`	`əndalɛñ`	`alsɛmmahumm.`
(4.131)	what	bring-IMP.2SG	that-he-said-to-me	I-didn’t-hear
	I didn’t hear what he told me to bring.			

(We use the orthography of [57].) The embedded clause is an indirect speech act derived from the command *bring that to me!* The adressee of the command is the speaker of the main clause, nevertheless it appears as 2nd person. The appearance of the word mən, meaning 'what' and not 'that' shows us that this is not direct speech. Here is some more data.

	aləttazzɛzɛzñalɛ.	
(4.132)	I-will-not-obey-me	he-said
	He refused to obey me.	

	alaggəzɛñ	alɛč.
(4.133)	I-will-not-help-me	she-said
	She refused to help me.	

	mɛskotu	aləkkɛffɛtəlləñ	alɛ.
(4.134)	window	I-will-not-open-for-you	said
	The window wouldn't open for me.		

What we observe is that the embedded clause contains 1st person agreement suffix referring to a 3rd person of the main clause, which happens to be the speaker of the subordinate speech act. So, Amharic indeed switches persons in an embedded speech act. However, notice that the verbs show double agreement, and each time with a 1st person (except for the last example above). As the glosses show, the first person subject refers to the speaker of the embedded speech act, while the first person object refers to the speaker of the main clause. We shall assume that in this case the 1st person is used deictically, to refer not to the speaker of the matrix clause, but in fact to the speaker of the utterance. Even though the two happen to be the same, this analysis would predict that intermediate speakers in double reported speech acts would be eligible for reference of a deictic pronoun. For example, it would predict that me cannot refer to Paul, nor Peter if used in this way. So, Amharic would in this case be different from English only in that he would be replaced by a first person pronoun. Moreover, this pronoun must refer to Peter and cannot refer to Paul.

(4.135) Paul said that Peter said that he has met me.

This analysis is quite in line with the previous analysis of temporal adverbials. Persons can be used absolutely, in which case they refer to the utterance situation, or relatively, in which case they obey the laws of succession.

Sequence of world effects appear as soon as we have conditionals. Consider the constrast between (4.136) and (4.137).

(4.136) `If ZFC proves its consistency, it is inconsistent.`

(4.137) `If you throw this piece of sugar into coffee, it dissolves.`

In (4.136) we find a material implication. Whatever the situation is, if the antededent holds, so does the consequent. We may however understand this in a stronger form, which is more readily seen in (4.137). In that reading it says that in any given world w, if the antecedent holds in w, so does the consequent in w. So, if the actual world is u, w can be any world. So we see here that the consequent picks up the parameter assignment of the implication. This actually seems to be a general feature of implication. The game is now played as above. We introduce parameters for the actual world and for the current world. By default, the current world is the actual world. A material conditional does not change that. A strict conditional however resets the current world (and puts this variable into the DRS head section in order to quantify over it). This is connected with morphology as follows. There are variants of the conditional (4.138) which assert that the current situation does not or cannot satisfy the antecedent. These are called the counterfactual conditionals.

(4.138) `If you had thrown this piece of sugar into your coffee, it would have dissolved.`

The semantics of the conditionals, especially the counterfactual conditionals, is a matter of its own. It would certainly go beyond the scope of this book to deal with it properly. Let us be content with noting just a few details that are independent of a proper formulation of the details. A conditional assumes the form $\varphi > \psi$, where φ and ψ are represented here as DRSs, and $>$ is a two place connective, forming a conditional DRS. In the simplest version, $>$ would simply be :, but there are numerous arguments against that view. φ and ψ share the same context, so all parameter values are set in the same way. (But see below.) However, the parameters of $\varphi > \psi$ as a whole are of course different. The counterfactual differs from the plain conditional in that the assertion $\neg\varphi$ (or even $\neg \diamond \varphi$) is also added, where $\diamond$ is simply a quantifier over possible worlds. (The reader may check that in the present framework, $\diamond$ is not needed, since we have explicitly stored the value of the world we are in. However, there is a limit as to how many iterations of $\diamond$ we can faithfully represent.)

Notice finally that conditionals can also change the values of the parameters in a rather subtle way. Take the following examples:

(4.139) `If I were you, I would not accept that offer.`

(4.140) `If I were you and you were me, I would not give you any money.`

Here, the antecedent resets the values of the person parameters. To see how this is represented in the present framework is a difficult affair. Certainly, it would not do to say in (4.139) that the speaker is equal to the hearer, since that would actually reset the assignment of the variables. Rather, we would have to assume that `I` and `you` in this sentence is crossidentified with the `I` of the current world and the `you` of the actual world. This is the only way of analysis, as (4.140) shows.

Notes on this section. The dependency of tenses, worlds and other coordinates, on the context, seems to be a problematic feature for Montague grammar. For we need to allow just about any word to use the current value of these coordinates. Our solution, based on parameters, allows to create a special list of variables or referents that are passed on in the construction of the semantic structure. This approach is quite awkward for Montague semantics, since we wouldn't know just how many of these coordinates there are, and each of them needs to be abstracted over individually. However, Schlenker [85] lumps all of them together into a single context variable. In this way, the semantics can be kept uniform. All that needs to be defined is what a context is.

Chapter 5

Case in Layers and Shells

We present a new theory of case. It assumes that cases are built up in layers, and with each layer the number of cases increases. The first layer is the morphological case, the second is the prepositional case taking an NP complement, while the third layer is a prepositional case taking a layer 1 PP. Languages have approximately the same number of cases according to our terminology. Case may be either selected by a head or not. We present a theory which allows for both to cooccur in a language. There are some languages, discussed at the end, which also allow for stacks of cases. Our analysis of these languages will generate an innovation in the system of variable handling by referent systems.

5.1 Syntactic and Morphological Cases

In Chapter 3 we have already spoken about case and case marking. In this chapter we will take a close look at the phenomenon. Moreover, we will discuss what cases are found in languages and how they function. We take here a very broad view as for what we call case. The most typical situation in which case appears is with a verb and its satellites. In dependency grammar one makes the following distinction. There exist (a) actants, and (b) circumstantials. Actants are roughly those satellites that the verb subcategorizes for in some way. Their case form is determined by the verb. Actants can be obligatory or non obligatory. Theories of syntax are mainly concerned with the case assignment of verbs to their actants. Here we will assume that case is everything that mediates between the NP satellite and the verb. In general, case is the mediator between a head and an adjunct or complement. If there was no case at all, an NP can just be put as it is in its position to yield a grammatical sentence. Hence, case is every morphological and syntactic material that is added to the bare NP. The following are therefore examples of case.

(5.1) `They wish him good luck.`

(5.2) `John never spoke about this incident.`

(5.3) `We believe in God.`

(5.4) `Larry threw the rubbish out of the window.`

(5.5) `A cat was climbing onto the roof.`

The fact that some cases carry some additional meaning (`among`, `from under`) is intended. The cases in question are the accusative in (5.1), the inessive in (5.3), the elative in (5.4), the sublative in (5.5). The case in (5.2) has no name as far as we know.

As will be seen in this chapter, case systems are rather involved even in languages that have no overt morphological case. For we will contend that the previous sharp distinction between morphological case marking on the one hand and syntactic case marking via functional words on the other is an illusion. A great many languages use a mixture of the two. Case is sometimes not assigned directly; it comes in layers. For example, in German we have four cases, nominative, accusative, dative and genitive. However, many words take arguments with a special preposition only. For example, (5.6) must be translated into German by (5.7) and (5.8) by (5.9).

(5.6) `He was thinking about this story.`

(5.7) `Er dachte über diese Geschichte nach.`

(5.8) `He was afraid of mice.`

(5.9) `Er hatte Angst vor Mäusen.`

So, the verb `nachdenken` (‘to think about’) must be construed with a complement opened by `über`; the noun `Angst` (‘fear’) must be construed with a complement opened by the preposition `vor` (‘in front of’). The cases in (5.6), (5.7) and (5.9) have no official names. The preposition `über` with accusative is typically used with content or topic, so that one might think of it as a case in itself. [1] Alternatively, we may think of it as being derived from a spatial case, which might be called the sublative II (see Section 5.3). The preposition `vor` is quite frequently selected by verbs in German and is perhaps therefore more case like than `über`. These prepositions must be learned, and there is regularity only up to a certain degree. In Hungarian, for example, one would rather say to have fear from something. So, (5.8) is translated by (5.5):

(5.10) `Félt az egerektöl.`
afraid-PAST.3.SG mouse-PL.ABL

We can see also that while English uses prepositions (and no cases), German uses a preposition plus a case. The preposition `über` ('over') assigns accusative case to its complement and the preposition `vor` assigns dative. The analysis we are pursuing here is to take the sequence of preposition and case assigned as one case, assigned in two layers (not shells!). These two together signal one case. In might be thought, however, that the assignment of morphological case is just a consequence of the assignment of the preposition, so all we need to know is which preposition is assigned to the complement by the verb and the morphological case is an automatic consequence. This however is not so. As we will see, many prepositions in German (and other Indo–European languages) assign several cases to their complement, depending on the meaning. Typically, there is a choice between dative and accusative. Dative must be used when there is no movement involved while accusative must be used when there is movement. Take for example the following sentences:

(5.11) `Er flog über die Berge.`
He fly-PAST.3.SG over the-ACC.PL.MASC mountains-ACC.PL

(5.12) `Er flog über den Bergen.`
He fly-PAST.3.SG over the-DAT.PL.MASC mountains-DAT.PL

(5.11) means that he was flying over the mountains from one end to the other. (5.12) means that he was flying, the mountains below him all the time. The only difference is the case on the nominal complement. Now go back to the example

[1] It is an accidental fact of German that the verbal prefix is separated from the verb in the main clause. So, to think about something is `nachdenken über etwas`, yet in a main clause the prefix `nach` is left at the end of the clause while the rest of the verb is put into second position.

(5.7). In the construction `nachdenken über` the complement has accusative case; to use the dative case yields an ungrammatical sentence.

(5.13) *`Er dachte über dieser Geschichte nach.`
He think-PAST.3.SG over this-DAT.SG.FEM story–DAT.SG

Before we discuss the technicalities of case we have to say something about the nature of selection. Typically, a verb is said to select a certain case for its complement. We want to advance the idea that the selection of a certain case is as much a function of the verb (and its meaning) as it is a function of the complement that is selected. Moreover, the more the construction fits to certain cases the more likely this case is actually chosen and thus the choice of the case is completely determined by the meaning of the expression. Let us illustrate this with an example. The German preposition `auf` (on) is used when something is on top of something else.

(5.14) `Albert sitzt auf einem Stuhl.`
Albert is sitting on a chair.

(5.15) `Albert frühstückt auf dem Dach.`
Albert is taking his breakfast on the roof.

In both examples, the preposition is used in its typical meaning. The same goes for `in` (in).

(5.16) `Karl ist in der Mine.`
Karl is in the mine.

(5.17) `Der Hund ist in seiner Hütte.`
The dog is its kennel.

When the verb is construed with abstract complements with a locative meaning, the distinction between being *on* or being *in* etc. become quite hazy. It is at this point that English and German start to differ with respect to the assignment of case.

(5.18) `Josef geht in die Schule.`
Josef go-3.SG into the-ACC school–ACC

(5.19) `Albert geht auf die Universität.`
Albert go-3.SG onto the-ACC university-ACC

It is difficult to say why we have to choose `in` when speaking about school or kindergarten and `auf` when speaking about the university or the high school. [2] Moreover, it can clearly not be attributed to the verb `gehen` that such and such case must be used. The concept expressed is in all these cases more or less the same; the only thing that changes is the character of the institution. Notice however that in all cases the fact that the verb `gehen` is a movement verb is reflected in the choice of the accusative for the complement. The accusative signals a movement towards the location (see Section 5.2). If we replace `gehen` by `sein` (to be) we get the following sentences.

(5.20) `Josef ist in der Schule.`
Josef is in the-DAT school-DAT.

(5.21) `Albert ist auf der Universität.`
Albert is at the-DAT university-DAT.

Here, the dative case must be chosen throughout. An additional twist in these examples is that in addition to the abstract meaning of the words school, university etc. there is a concrete one, namely the building itself. In this case, the choice of the case is determined by the way in which the person is located. Of course, this may lead to ambiguities. (5.20) can mean both that Josef goes to school and that he is in the school building.

The problems just illustrated also arises with concrete objects. For example, to be in the garden or the field is expressed differently in German.

(5.22) `Albert ist im Garten.`
Albert is in the garden.

(5.23) `Albert ist auf dem Feld.`
Albert is on the field.

To give a more striking example, with respect to a city or country, do we want to say that we are *in* it or that we are *on* it? The answers are again different in different languages. English and German agree to use *in*. [3] In Hungarian, there is a case expressing the meaning *in* (the inessive) and one case expressing

[2] To be exact, there is a choice between three prepositions: `zu`, `auf` and `in`. `Kindergarten` can only be construed with `in`, `Schule` allows all three, `Gymnasium` only `in` and `auf` and `Universität` allows `zu` and `auf`.

[3] But notice the choice of the preposition in the directional use.

the meaning *on* (the superessive). Which of the two is appropriate is difficult to predict. Foreign places are usually construed with the inessive while Hungarian place typically (but not always) take the superessive. We have for example

(5.24) `Berlinben vagyok.`
Berlin-INESS be-1.SG

(5.25) `Szegeden vagyok.`
Szeged-SUPESS be-1.SG

Place names ending in `falu` ('village') select the inessive, and place names ending in `falva` ('village') select the superessive. The determining factor may in these cases well be phonological.

We finish with another observation concerning the use of directional cases and non directional cases. We have seen that the motion towards a location in German is usually distinguished from the being at the location only by the contrast between accusative and dative, while the preposition remains the same. It may therefore appear that movement verbs select a preposition with the accusative (if the motion is indeed towards the object) while stative verbs select that preposition plus the dative. However, this again depends on various factors. First of all, even when something is moving the location in which it moves may be the same throughout the event. Therefore, the following are grammatical sentences.

(5.26) `Egon geht auf das Dach.`
Egon goes onto the-ACC roof-ACC

(5.27) `Egon geht auf dem Dach.`
Egon walks on the-DAT roof-DAT

So, a movement verb may tolerate a stative case. However, verbs that do not express a movement (or change) may also tolerate directional cases.

(5.28) `Egon säuft sich noch ins Grab.`
Egon will drink himself into the grave.

(i) `Ich bin in Berlin.`

(i′) *I am in Berlin.*

(ii) `Ich gehe nach Berlin.`

(ii′) *I go to Berlin.*

To use `into` in (ii′) is possible but has a special meaning.

The use of directional and non directional again becomes arbitrarily fixed by the verb once the complement has no locational meaning but is rather abstract. Consider once again the examples (5.6) and (5.7). *Think* is not directional, yet the complement must be construed with accusative, not dative. There is no choice; the dative is ungrammatical here.

Notes on this section. Our view on cases is markedly different from the one dominant in GB and subsequent theories. There, only two cases were seriously discussed, namely nominative and accusative, while the remainder of cases, called 'inherent', was seen as outside the scope of syntax. Moreover, as all central notions were defined in terms of structure, it made a big difference whether a language realizes a case morphologically or not. In the latter case, the argument is a PP rather than a DP, which meant that the DP complement of the P is unable to bind outside of the PP. This applies even to dative complements in English! (See [89] for a discussion.) Under this view, one would expect for a language in which every case is realized by a preposition, for example Cebuano, that binding theory is trivial, since a DP is always preceded by a preposition. Hence, no reflexivization would be possible. This is not the case, see [11]. It seems, then, that the distinction between morphological cases and nonmorphological cases is syntactically less relevant than hitherto assumed. Webelhuth [97], working within GB, comes on Page 141 to a similar conclusion. He shows that Ps pattern in many respects with inflectional affixes and explains this on the basis that prepositions pass on θ–roles. Thus, rather than assigning a θ–role by themselves, they simply pass on the one assigned by the verb to the DP. This comes close to our conception of prepositions as case mediators (which includes mediation of θ–roles in an indirect way as well). Finally, as will be seen, they also do not simply assign some case to their complement, but once again there is an influence of the higher verb on the case (for example, accusative versus dative with locative complements in German). The way this is resolved here is by assuming a case assignment in layers, as outlined in Section 5.4.

5.2 An Overview of Cases and Case Functions

The term 'case' usually refers to distinct (case-)forms in a language. A **case function** on the other hand is a basic meaning or construction type of a case. Ideally, a language should have a number of case forms and a mapping from case functions to case forms, which let us predict which case form to put under what circumstance. It is however often hard to say in what function a certain complement appears, and we shall see that languages tend to either make a specific choice or leave the matter open. The discussion below does not enumerate cases as such but rather case functions. We know of no language in which the case functions are as clearly separated as they are below, which is also due to the fact that very often clear divisions simply cannot be made. The reader will also meanwhile have noticed that our use of case encompasses far more than the morphological cases. Roughly, we speak of a case whenever the particular case function is subcategorized for by some word. This has the following reason. If some lexical element subcategorizes for a certain case(-function) it typically makes the case devoid of any meaning. We have seen such examples in the previous section. We have to use the German verb `denken` with the locative expression `an` ('at'). This is a more or less arbitrary choice of the German language; it could have been otherwise. For example, there exists a transitive verb `bedenken` ('consider'), selecting the accusative. The locative meaning of `an` is lost in connection with the verb `denken`. Therefore we say that the preposition `an` together with the accusative that it selects in turn, is a case.

Structural Cases The notion of a structural case is reserved for those cases which carry no meaning by themselves but are used only to distinguish the arguments of a word (typically a verb) from each other. In Government and Binding theory, two cases are assumed to be structural, namely the nominative and the accusative. However, there is a number of facts that militate against this view. First of all, verbs take up to three arguments (see for example [20]). These arguments are distinguished in some languages by three distinct cases. Moreover, the assignment of cases is arguably arbitrary, at least with respect to the cases different from the nominative. We will call the basic structural cases **nominative**, **accusative** and dative, following traditional usage. The nominative is typically reserved for the most actor–like argument. The accusative is typically reserved for the direct object, though it is hard to define the notion of a direct object without taking recourse to the accusative. Suffice here to say that the accusative argument is normally privileged in various ways. Languages with double agreement tend to have agreement with nominative and accusative arguments (for example Hungarian). Accusative objects are in many languages the only ones that can undergo passive.

Associating Cases A case is associative if it stresses the relationship with another thing. The most common case is the **genitive**. The genitive expresses possession in the widest sense. It is often difficult to distinguish possession from other forms of relatedness. Some languages further distinguish **inalienable possession** from **alienable possession**. Body parts are typically inalienable; one cannot give them away except under special circumstances. Relatives might belong to that category. Alienable things are those which we intuitive classify as 'possessed'. If something is not alienable it is not possessed. I do not possess my hand, it is simply my hand. Again, these are matters of arbitrary decision and vary from language to language. The loosest relation between two nouns is expressed in the **comitative**. Typical uses of the comitative are sentences such as

(5.29) `John came with his wife.`

(5.30) `Pete saw John with his wife.`

The comitative is often hard to separate from the adessive (see below), which is typically expressed by the preposition `near`. The latter is a locative case expressing that something is near in location to something else. (5.29) and (5.30) represent clear uses of the comitative. For even though John's wife was near him when he came, the adessive is not appropriate here. For the adessive does not tolerate any movement. It expresses that something was near to something else throughout the whole event. Moreover, there is an element of volition or intention involved which sometimes is visible in the way the two people interact when they are together as opposed to being simply near each other. A very clear case is the distinction between `dance with someone` and `dance near someone`.

In addition, the intentional element in the comitative implies a certain overlap with the instrumental (see below); the fact that in (5.29) John was together with his wife is not accidental. If he happened to come at the same time as his wife, this does not license the use of the comitative (5.29). There must be an intention on John's part to be with his wife. Likewise (5.30). For Pete to see John with his wife means that he saw them not simply near each other but clearly being together (involving the fact that they knew they were near each other etc.). The instrumental expresses that something (the instrument) is the means of achieving something else. Typically, an instrument is near the person who uses it. Therefore, the instrumental carries a meaning of togetherness and in certain languages the instrumental is indeed not separated syntactically (or morphologically) from the associative. Examples are English (`with something` means both together with something and using something) and German (`mit etwas` likewise means together with something and using something).

Some language also have a so-called *abessive* (sometimes called *caritive*). It expresses that something is absent. In English, the abessive is expressed by the preposition `without`.

(5.31) `He went into vacation without money.`

Intentional Cases The cases that express intentions are the **instrumental**, the **benefactive** and the **finalis**. We also group with them the **causative**, because it shows great overlap with the finalis. The instrumental and the benefactive are by far more frequent that the other two, which are usually expressed not by means of a nominal group but by means of a verbal complement. The instrumental is used to express the means by which something is achieved. In English, the preposition `with` is used to express the instrument.

(5.32) `He cut the cake with a knife.`

(5.33) `He smashed the window with a stone.`

(5.34) `They destroyed the house with dynamite.`

To use something as an instrument normally requires an intention. However, it is alright to say (5.33) even when he smashed the window only accidentally, say by throwing the stone somewhere. In this case he only intended to throw the stone while by doing that the window got smashed. We have also said earlier that the instrumental is close to the comitative, since using certain everyday instruments (hammer, knife, screwdriver etc.) we need to be together with that instrument in a certain canonical way in order to use it. However, this is only a coincidence. In (5.34) we certainly do not want to be together with the dynamite when it destroys the house.

The benefactive expresses the person or thing for which the action is intended. The most typical action through which someone benefits is the act of giving; this is why the benefactive is often called *dative*. Here are some example.

(5.35) `John gave Susan an apple.`

(5.36) `John cooked a soup for Susan.`

(5.37) `John bought Susan a car.`

Notice that despite the name the benefactive may also be used when that person is badly affected. (Some therefore use the term **malefactive**.)

The causative is used when something is the reason for an action, as in the examples (5.38) and (5.39).

(5.38) `Because of John we came late.`

(5.39) `Through John we became acquainted with the dean.`

The finalis expresses the goal that should be achieved through an action. It resembles the causative in that the goal is usually the cause that makes us do something. In English the finalis is expressed by `in order to` or simply `to`, but has a verbal complement. Occasionally, one can use also `for` (plus nominal complement).

(5.40) `I am going to buy bread.`

(5.41) `I am going for the bread.`

The Partitive The **partitive** forms a class of its own. Usually the partitive is in contrast with the accusative of transitive verbs or the nominative in intransitives. The partitive denotes the fact that the object is only partially affected or that there is some unspecified amount of that object. English has no partitive. The morphological partitive is found — among other — in Finnish and Russian. Consider the following contrasting sentences of Finnish.

(5.42) `Hän syö kalaa.`
$He eat--3.$SG$fish--$PART.SG.

(5.43) `Hän syö kalan.`
$He eat--3.$SG$the fish--$ACC.SG.

The partitive is a rather delicate case. The choice between partitive and direct case is usually the result of many factors, such as aspect of the verb, force, and polarity among other. The negative usually cooccurs with the partitive. The partitive is expressed in Hungarian using the elative and in German using the Ablative.

(5.44) `Er hat davon gegessen.`
he has there–from eaten

(5.45) `Ettől evett.`
this–ELA eat–3.Sg.Past

Locatives The locatives form not only a rich class of cases but also one that is highly structured. Locative prepositions are of the form $M + C$, where C is an element taking a DP and forming a location (called **configurator**) and M an element taking a location and forming an adverbial expression (called **modalizer**). Configuration and mode are the two axes along which locatives are classified, although we shall see that locative cases may not be analyzed as feature matrices, say of the form

(5.46) $$\begin{bmatrix} \text{CONF} & : & \kappa \\ \text{MODE} & : & \mu \end{bmatrix}$$

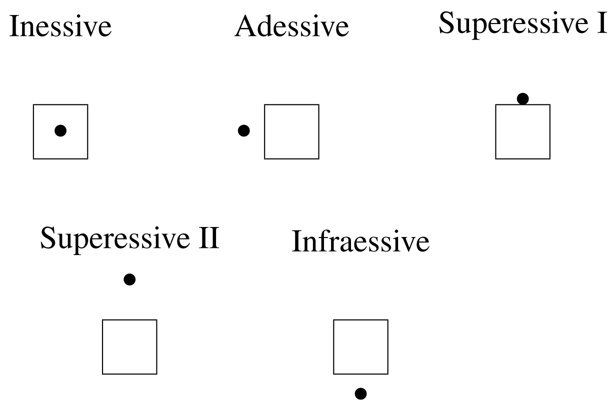
The mode defines the type of movement (or absence of movement) and the configuration concerns the location relative to the complement noun phrase (called the **land mark**). Let us start with the latter. We call a (spatial) configuration of some objects a constellation of the object lying with respect to each other. Typically, cases are configurations involving two objects. There is a multitude of configurations, and as it will turn out not all of them are apply in all cases. One can systematize the configurations of two objects according to two parameters. The first is the direction of the arrow point from the one to the other. We call it the **direction**. The other is the **distance**. Typically, with respect to distance it is distinguished between (a) touching, (b) near, (c) not near. With respect to direction for two objects we have (1) next to, (2) in, (3) above, (4) under, (5) facing. Some languages also indicate whether the object is to the north, south etc. of the other. But this is the minority of cases. The most basic configurations are to be *in* (**inessive**), to be *at* (**adessive**, glossed as ADESS), to be *on top* (**superessive I**, glossed as SUP I), to be *over* (**superessive II**, glossed as SUP II) and to be *under* (**infraessive**, glossed as INFESS) some object (see Figure 5.1). To be in something is not qualified with respect to distance. To be at something means to be at least near ((a) or (b)). To be on top means above and touching, to be over something means above and not touching. To be under something may be any of the three. There is no distinction between an infraessive I (touching) and infraessive II (non touching). We know of no language in which this distinction is syntactically relevant. The configuration of the infraessive is simply neutral with respect to the closeness of the two objects. The following sentences exemplify these different concepts.

(5.47) `The nail is in the box.`(inessive)

(5.48) `The guests are at the door.`(adessive)

(5.49) `The clock is on the radio.`(superessive I)

Figure 5.1: The Configurations



(5.50) `The airplane is over London.`(superessive II)

(5.51) `The key is under the mat.`(infraessive)

An additional configuration is **betweenness**. It is to my knowledge not realized as a morphological case. It is realised in English through the prepositions `between` and `among`. Between is a relation between a single object and a group of at least two objects.

Somewhat more involved are the spatial configurations that use intrinsic orientation of the objects. These are represented by `in front of` or `facing` and `behind`. Their use is quite delicate. German `vor` ('in front of') is used in two meanings. One can say that x is `vor` y when the object x is either (i) between you and y or (ii) facing y.

This closes the list of spatial configurations. Now we turn to what we call the **modes**. We distinguish **static** from **nonstatic** modes. A mode is static if the configuration is stable throughout the event. Otherwise it is nonstatic. Above we have used only static modes. In a nonstatic mode we have to distinguish several ways of change. Take the following example.

(5.52) `John is in the house.`

There are various ways in which the situation depicted in the sentence can occur in event. Either it holds at the beginning of the event (*coinitial*) or it holds at the

Figure 5.2: The Nonstatic Modes

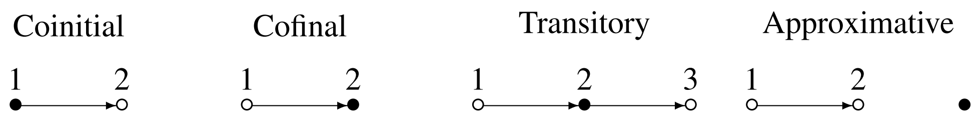


Table 5.1: The English Locatives

	stative	coinitial	cofinal	transitory
inessive	`in`	`out of`	`into`	`through`
adessive	`at`	`from`	`to`	`through`
superessive I	`on`	`from`	`on(to)`	—
superessive II	`above`	`from above`	—	`over`
infraessive	`under`	`from under`	`under`	`under`

end (**cofinal**) or it holds at some time in between (**transitory**).

(5.53) `John went out of the house.`

(5.54) `John went into the house.`

(5.55) `John went through the house.`

A fourth mode that is attested in some languages is the **approximative**. The mode is approximative if the object moves towards the location during event time. The transitory and the approximative modes are not so frequent as the others. Also, the transitory mode has usually a special meaning. To move through the house is not simply to go in and then out of it. It means to enter at one door and to exit at some door opposite. A better example is `through the tunnel`. Since a tunnel has mostly two entrances, to go through a tunnel means to enter at one and to leave it at the other. A final note concerning the terminology. The stative cases are usually named by a construction of the form *preposition+essive*. We have followed that usage. The other cases are usually named by a construction of the form *preposition+lative*. For example, *illative* is the cofinal counterpart of the inessive. In grammars, only the Inessive, Adessive and Superessive I have a full series of counterparts, listed in Table 5.2. The others are to our knowledge never

Table 5.2: Locative Case Names

Stative	Coinitial	Cofinal	Transitory
Inessive (INESS)	Elative (ELA)	Illative (ILL)	Perlative (PERL)
Adessive (ADESS)	Ablative (ABL)	Allative (ALL)	Translative (TRSL)
Superessive (I) (SUP (I))	Delative DEL	Sublative (SUBL)	? ?

morphologically institutionalized.

Temporal Locatives Just like two objects are in a certain location with respect to each other, so can be events with respect to their time. Things can happen *at* some time, *before* some time, or *after*. What distinguishes time from location, however, is that it is one–dimensional and directed. Nevertheless, the list we have just given is not exhaustive. We can classify temporal locatives just the same way as locatives. There is a configuration and there is a mode. We have already mentioned three modes: anterior, cotemporaneous and posterior. We have all four modes. An event is **static** with respect to a configuration if it occurs entirely within that configuration; it is coinitial if it starts in that configuration; it is cofinal if it ends up in the configuration, and it is transitory if in enters and then leaves the configuration. This analogy is the reason why the temporal locatives are usually replaced by their spatial locative counterpart. However, notice that while the modes are completely analogous, the configurations are not. We will not go into the details of temporal cases but notice that in Hungarian there exists a special suffix for time. The temporal locatives have a special mode, namely the *iterative*. The iterative appears also on verbs, but Hungarian actually has a special case reserved for iterative events. We can express that an event happens *every X* by using a special suffix (`-nta`).

Qualitatives A typical qualitative case is the *essive*. It is used to express a property the subject (or salient nominal) has. It occurs in English in the expression `as a NP`. We can say that the subject is actually *in* the quality. This allows to draw

Table 5.3: Some Chinese Cases

PREPOSITION	VERB MEANING	CASE FUNCTION
bǎ	*grasp*	accusative
gěi	*give*	dative, beneficiary
yòng	*use*	instrumental
ná	*take*	instrumental
gēn	*follow*	comitative
zài	*be present*	locative
cóng	*follow*	ablative
dào	*arrive*	terminal

a useful analogy with locatives, since qualitative cases also have four modes, but only one configuration. Typical other qualitatives are the translative of Finnish or the transformative of Hungarian. They correspond to the cofinal mode. An English equivalent would be `turning into NP`.

5.3 An Outline of Some Case Systems

Chinese Chinese has no morphological inflection. There are a number of prepositions that function like cases in the Indo–European sense. These prepositions are verbs, just like in many Creole languages. In [72], page 163, the following correspondence is given between cases and prepositions, shown in Table 5.3.

(5.56) `Tā cóng Bólín lái de.`
He ABL Berlin come PERF
He comes from Berlin.

(5.57) `Tā gěi wǒ jièshào Wáng xiānshéng.`
He DAT me introduce Wang Mr.
He introduces Mr. Wang to me.

There are no functional words acting as intermediaries (as in Hindi). Furthermore, there are so–called split prepositions. Examples are

Table 5.4: Core English Cases

	SG	PL	SG	PL
NOM	the car	the cars	I	we
GEN	the car's	of the cars	my	our
			of me	of us
DAT	to the car	to the cars	to me	to us
ACC	the car	the cars	me	us

zài ...	li	in
zài ...	xia	under
zài ...	shang	on

Since zài is a preposition (locative), we conclude that the other element is an optional qualification.

English We have dealt previously with English examples. Here we will concerns ourselves only with the question whether English has morphological cases at all. If we only look at proper nouns we find only a nominative and possibly a genitive. However, the genitive is realized through a clitic 's and therefore not a morphological case, which is clear when we look at the plural. However, the English pronoun system shows a distinction between nominative, genitive and accusative, as is shown in Table 5.4. The morphological genitive is reserved exclusively for the possessive function. There is also the genitive by means of of; this is the one that can be selected.

(5.58) It was kind of him to give us a lift.

(5.59) *It was kind his to give us a lift.

Prepositions take the accusative throughout. The nominative is reserved for the subjects of a sentence. In practical syntax one would in fact prefer to say that English has two genitive cases, because they are not always freely interchangeable.

French French is similar to English in having no morphological case on nouns, but a somewhat more elaborate system of pronouns. However, the details are

Table 5.5: Core French Cases

	SG	PL	SG	PL
NOM	la valise	les valises	il	ils
GEN	de la valise	des valises	son	leur
			de lui	de eux
DAT	à la valise	aux valises	lui	eux
ACC	la valise	les valises	le	les

nevertheless different. The pronouns show a threefold distinction between nominative, dative and accusative case. The genitive unites the possessive, the partitive, and is also a structural case. In addition, it is used for coinitial locatives, mainly the *elative* and the *ablative*. [4]

(5.60) `C'est la voiture de Jean.`
This is the car of John.

(5.61) `Il (ne) mange (pas) de pommes.`
He eats (does not eat) some apples.

(5.62) `Jean se souvient de ses vacances.`
John remembers his vacation.

(5.63) `Jean vient de la gare.`
John is coming from the train station.

Likewise, à not only is a structural case, and expresses the benefactive, it is also used in many locative senses, mainly stative and cofinal.

(5.64) `Jean donne un chat à Marie.`
John gives a cat to Mary.

(5.65) `Jean est à Paris.`
John is in Paris.

(5.66) `Jean se promène au jardin.`
John is walking to the garden.

The meaning of de and à is therefore quite abstract, see [15].

[4] In French, de+les is realized as des, à+le as au and à+les as aux.

Table 5.6: German Locatives

	stative	cofinal	coinitial	transitory
adessive	an + DAT	an + ACC	von + DAT	an + DAT vorbei
inessive	in + DAT	in + ACC	aus + DAT heraus	durch + ACC
superessive I	auf + DAT	auf + ACC	von + DAT weg	?
superessive II	über + DAT	bis über + ACC	?	über + ACC (hinweg)
infraessive	unter + DAT	unter + ACC	unter + DAT hervor	unter + ACC durch

German German has four morphological cases, nominative, accusative, dative and genitive (see the paradigm in Section 3.3). It has both prepositions and postpositions. The overwhelming majority of prepositions select the accusative and dative, only a small number the genitive (for example entgegen, and wegen (however only for older speakers)). The postpositions, however, mostly select the genitive, some the accusative (entlang, hinab) and some the dative (nach).

As we have outlined above, many prepositions may alternatively select the accusative or the dative. The choice is then determined by the meaning. If the meaning is static, then the dative is chosen, if it is coterminal, the accusative is chosen.

Finnish Finnish has six locative cases, corresponding to the inessive and adessive configuration, using stative, cofinal and coinitial mode. Moreover, there is a nominative, a partitive, an essive, a translative, an abessive, a comitative and an instructive. (See Karlsson [52] for details.) The accusative is claimed not to be a genuine morphological case. Table 5.7 shows the cases. The plural is regularly formed by using an infix (-i-). The name **translative** (glossed TRSL) is actually not such a good term. The case is used for the state one transforms into.

(5.67) Isä on tullut vanha-ksi.

Table 5.7: The Morphological Cases of Finnish: `talo` (*house*)

NOM	talo	ESS	talona
PART	taloa	TRSL	taloksi
GEN/ACC	talon	ABESS	talotta
INESS	talossa	ADESS	talolla
ILL	taloon	ALL	talolle
ELA	talosta	ABL	talolta

father is become old–TRSL
Father has become old.

Therefore, the translative is related to the essive, which expresses a state in which one already is.

(5.68) `Heikki on Jämsä-ssä lääkäri-nä.`
Heikki is Jämsä-INESS doctor-ESS
Heikki is a doctor in Jämsä.

The instructive is actually an instrumental, but it is used only in certain expressions. The comitative is obligatorily used together with a possessive suffix. Moreover, there is no distinction between a singular and a plural comitative.

(5.69) `Rauma on mukava kaupunki vanho-ine talo-ine-en.`
Rauma is beautiful-NOM.SG city-NOM.SG old-COM house-COM-POSS:3
Rauma is a beautiful city with its old houses.

The adessive of Finnish collects different case functions: the adessive, the superessive I, the possessive and the instrument.

(5.70) `Vainikkala on Neuvostoliito-n raja-lla.`
Vainikkala-NOM be.3.SG.PRES Russia-GEN border-ADESS

(5.71) `Antti on laiva-lla.`
Antti-NOM be.3.SG.PRES ship-ADESS

(5.72) `Minu-lla ei ole raha-a.`
I-ADESS not.3.SG be money-PART

Table 5.8: The Morphological Cases of Hungarian

NOM	ház	INESS	házban
ACC	házat	ELA	házból
BEN	háznak	ILL	házba
COM	házostul	ADESS	háznál
FIN	házért	ABL	háztól
FOR	házként	ALL	házhoz
TRSF	házzá	SUP I	házon
INST	házzal	DEL I	házról
TERM	házig	SUB I	házra

(5.73) Syö-n keitto-a lusika-lla.
Eat-1.SG.PRES soup-PART spoon-ADESS

There are other uses of the adessive (for example temporal uses), which we will not go into. We note that just like the adessive, the Finnish ablative collects the ablative and the delative function, and the Finnish allative the two functions of allative and the sublative.

Hungarian The morphological cases of Hungarian are listed in Table 5.8. The plural is regularly formed by adding the plural suffix to the stem. For example, the plural of ház is házak. We left out two cases. These are temporal locatives. They are called the *temporal* and the *distributive temporal*. The first is used to express that something takes place at a certain time point, the second that it takes place regularly at a certain time point.

(5.74) öt órá--ig = at five o'clock

(5.75) csütörtök--önte = every Thursday

(5.76) hétfő--nte = every Monday

The transformative (glossed TRSF) of Hungarian is actually equivalent to the Translative of Finnish. The different names result from a different tradition in nomenclature in the two philologies. We have pointed out earlier that the name *Translative* is actually not so suggestive.

We can see that Hungarian adds to the Finnish locatives another configuration. It differentiates in contrast to Finnish the adessive and the superessive I configuration. There is also a peculiarity of the Hungarian appositions that is worth mentioning. First of all, all appositions are postpositions; moreover, they govern the nominative case. Since the nominative has a zero suffix, it is quite hard to distinguish between a postposition and a case suffix. What is more, postpositions participate in the tripartite mode–system. Consider, for example, the configuration of the infraessive. It is perfectly conceivable to add any of the four modes to the infraessive. However, hardly any language cases corresponding to these functions. What is more, appositions are provided mostly for the stative mode, and the other modes must be provided by other means. For in example, in English we can say *Jack is under the table.* but to use the the coinitial mode we need to say something like `Jack emerged from under the table.` So, `from under` is the complex that signals the infraessive in coinitial mode. Hungarian, by contrast, provides a set of three postpositions corresponding to the stative, cofinal, and coinitial mode.

(5.77) `A cica az asztal alatt.`
the cat the table under-STAT

(5.78) `A cica az asztal alá fut.`
the cat the table under-COFIN

(5.79) `A cica az asztal alól jön.`
the cat the table under-COINIT come-3.SG.PRES

That this is a regular feature of the Hungarian appositional system is shown in Table 5.9.

Hindi Hindi has two cases, **direct** (glossed DCT) and oblique (glossed OBL).

	singular	plural
DCT	kamrā	kamre
OBL	kamre	kamrom̃

There is a small class of so–called *primary postpositions*. These are `kā` for the genitive, `se`, for the instrumental–ablative, `ko`, for the dative, and `mẽ`, for the locative (see [67] and [65], page 233). They select the oblique case and function

Table 5.9: Hungarian Locative Postpositions

	static	cofinal	coinitial
under	alatt	alá	alól
above	fölött	fölé	fölül
next to	mellett	mellé	mellől
in front of	előtt	elé	elől
behind	mögött	mögé	mögül
among	között	közé	közül

like clitics. In some Indo–Aryan languages (Marathi) they have already become suffixes, so that we find the not so unusual pattern that certain cases are built using a different stem (the oblique stem). Such is the case, for example, in Latin.

(5.80) us ādmī ko tīn pustkem̱ dījie
that.OBL man-OBL DAT three book-PL.OBL give-IMP
Please give that man three books.

(5.81) us-se yah savāl pūchie
that.OBL-SE this-DCT question-DCT ask-IMP
Please ask him this question.

In addition to these postpositions there is a much wider class of postpositions that typically select a combination of noun phrase and primary postposition. Here are some examples.

(5.82) laṛke ke sāth
boy-OBL GEN-OBL with
with the boy

(5.83) makān ke pīche
house-OBL GEN-OBL behind
behind the house

It is to be noted that the primary postpositions also inflect for direct and oblique case.

Table 5.10: The Locatives of Avar

	Modus →			
Configuration ↓	Stative	Coinitial	Cofinal	Transitory
On	-da	-d-e	-da-ssa	-da-ssa-n
At	-q	-q-e	-q-a	-q-a-n
Under	-λ'	-λ'-e	-λ'-a	-λ'-a-n
In	-λ	-λ-e	-λ-a	-λ-a-n
In a Hollow	-∅	-∅-e	-∅-ssa	-∅-ssa-n

Avar Many Caucasian languages have a rather full set of locatives. A particularly striking case is Avar. The data is taken from [13]. There are 27 cases, of which 20 are locative cases. They are summarized in Table 5.10. What can be observed is that the cases consist of up to three suffixes; first suffix characterizes the configuration, the other two the modus. It appears that the transitory modus is derived from the cofinal modus by the suffix -n. Tsez, another Caucasian language, has in distinction to Avar no transitory mode but an approximative instead (see [19]).

5.4 Layers of Case

The idea of layers of case is taken from Masica [65]. It is argued there that Indo–Aryan languages possess up to four layers of case. The first layer consists of what one would normally call case; it is typically reduced to a distinction between direct and oblique (as in Hindi). This is the morphological case. All other case layers consist in independent words. The postposition ko, for example, is used for the dative function. It selects the oblique case. The elements of Layer II may also change according to number and gender, that is, they show in some cases agreement with the complement. An example of an element of Layer III is sāth (*with*), which takes a genitive complement. So we have

	laṛke	ke	sāth
(5.84)	boy-OBL	kā-OBL.MASC	with
	noun.I	II	III

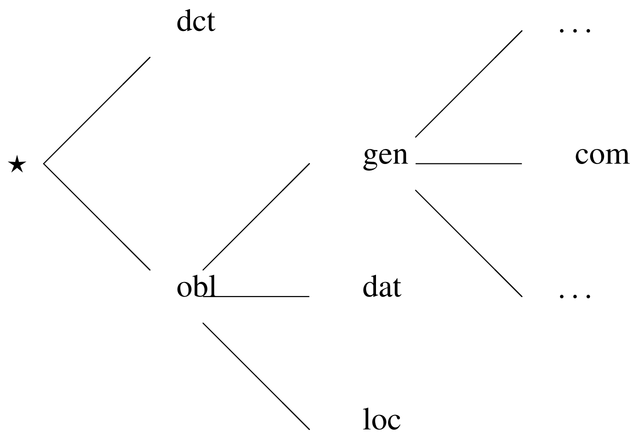
There is also talk of a fourth layer of cases, though the evidence is less stringent.

Various things need to be made clear. First, there is no intrinsic reason for eg the dative to be of Layer II. For example in Kashmiri there is a dative of Layer I. Hence, the layers are defined only relative to each other and within one language. What is of Layer II depends on whether it selects a phrase of Layer I. The second thing is that what is called case here is not a case in the traditional sense, but we have adopted that terminology for our purposes. Third, what the notion of case suggests here is typically covered by the term 'case function'. However, there are differences. For given a verb together with its meaning a particular actant can have only one case function. (If it has two case functions, the verb is actually ambiguous.) However, as we will see below, there exist verbs in different languages (with arguably identical meaning) that select different cases (for example sublative versus illative versus dative). This is not so because either of the languages lacks one of these cases. Rather, the cases are there, but the verbs chooses one as opposed to the other. Hence, the cases that we define are not case functions, they stand in the same relationship with case functions as the traditional cases. It might therefore still be the case that a language lacks a particular case. A case in point is the dative in Finnish. The dative is absent and its use is taken up by other cases (eg the allative). Hence Finnish has even in the enlarged sense of case no dative. English, however, has a dative. It is expressed by the Layer II preposition `to`, which in turn selects accusative. Our definition of the cases besets the same problem as with the morphological cases. Although they are distinct from case functions, they are nevertheless named after the primary case function that they serve.

It is apparent that the examples above are no different from what we find in other languages around the world. So, the notion of case in layers may be universally applied to just about any language. The cases of a language are like a fan, which grows larger and larger the more layers are added. This fan is illustrated for Hindi in Figure 5.3.

The potential use of the idea that languages have the case spread out in layers is that one will be less concerned with the number of cases, since this will turn out to be more or less the same number across languages whereas the number of morphological cases can vary greatly, but it will also show that the selection of prepositions is no different in principle from selection of case. Moreover, it will turn out that even morphologically impoverished languages nevertheless have a rich case system, though obviously not realized through large morphological

Figure 5.3: The Hindi Case Fan



paradigms. We will illustrate our point with German. German has four cases in the first layer. There is agreement with the adjective in case of the first layer. There are numerous words (verbs, nouns, adjectives and adverbs) that do not select a noun phrase in a particular case but select a prepositional phrase. The preposition in turn selects a noun phrase in a particular case.

Let us take an example. The verb `sich beziehen` (*to refer*) takes as a complement a PP with the preposition `auf`. Moreover, the case governed by `auf` must be the accusative. The use of the dative would be ungrammatical. It is customary to refer to the name of the preposition and the case it in turn governs in the subcategorization frame.

`sich beziehen` [__, `auf` + ACC]

Here we shall take a different approach. We shall say that what gets realized as `auf`+ACC is actually a case in its own right, namely the sublative. So, `sich beziehen` does select a case, namely the sublative. Of course, German has no morphological case named 'sublative'. Rather, it has a lexical entry, `auf`, that can change the accusative case into the sublative. In this way the case assignment is mediated between the verb and the noun phrase.

(5.85)

$$\begin{array}{|c|}\multicolumn{1}{c}{/\texttt{auf}/} \\ \hline \langle x : \Diamond\ogreaterthan : [\textsc{case} : acc] \mapsto [\textsc{case} : subl]\rangle \\ \hline \varnothing \\ \hline \varnothing \\ \hline \end{array}$$

Clearly, `auf` is a Layer II sign. Morphologically, `auf` is a phrasal affix. This means that it may not be separated from the phrase to which it attaches. This has several consequences. In German, there is no preposition stranding, for example. (English is a different matter. Generally, what seems to be the case is that a verb can incorporate a preposition which then looses its status as a phrasal affix. Therefore it appears as if English would violate the ban on separation of prepositions and complement. English however obeys the latter restriction but has optional incorporation. Since the limit of incorporation is that of a single argument, we should expect that English does not allow double preposition stranding. This is the case.) Notice that the semantic contribution of `auf` is zero. This is so because it is the verb that decides what it will do with the meaning of the noun phrase. The case selection is (within bounds) idiosyncratic. In Hungarian, the corresponding verb, namely `vonatkozik`, also selects the sublative, the Finnish `viitata` takes illative. Notice that `to refer` in English takes dative case. So, what distinguishes these items is not the syntactic structure they project but rather only the case they select. If we nominalize the verb the corresponding noun selects the same case. We have `der Bezug` (*the reference*), which selects the sublative. If we turn this into an adjective (`bezüglich`, *with reference to*), it too selects the sublative. As a rule, the selection of cases other than nominative, accusative and dative remains stable under change of category. We therefore see that selection of more complex cases is independent of the category of the head.

For Hindi, we have the following lexical structures:

(5.86)

$$\begin{array}{|c|}\multicolumn{1}{c}{/\texttt{ke}/} \\ \hline \langle x : \Diamond\olessthan : \left[\begin{array}{lll} \textsc{case} & : & obl \mapsto gen \\ \textsc{num} & : & pl \\ \textsc{class} & : & masc \end{array}\right]\rangle \\ \hline \varnothing \\ \hline \varnothing \\ \hline \end{array}$$

/sāth/

(5.87)

$\langle x : \Diamond\circleddash : [\textsc{case} : \mathit{gen} \mapsto \mathit{com}]\rangle$
∅
∅

A general fact about cases is that agreement in case within a noun phrase goes only up to a certain layer. Typically, it stops at the first layer. When there is no agreement within the noun phrase we also speak of a group marking language (eg Hungarian). However, when there is agreement, then we speak of a word marking language. Agreement apparently never extends beyond the first layer of case:

(5.88) `Ich beziehe mich auf den großen Roman von Tolstoj.`
I refer-PRES 1.SG.REFL ALL the-ACC great-ACC novel-ACC by Tolstoy
I am referring to the great novel by Tolstoy.

(5.89) *`Ich beziehe mich auf den auf großen auf Roman von Tolstoj.`

The reason for this behaviour is that `auf` takes a phrasal complement. However, since there is no explicit notion of a phrase in this framework, we shall say that `auf` selects a nominal complement with definiteness feature defined. In this way it is made sure that it is a preposition taking to its right an accusative complement. The noun phrase shows accusative agreement internally.

/auf/

(5.90)

$\langle x : \Diamond\circleddash : \begin{bmatrix} \textsc{case} & : & \mathit{acc} \mapsto \mathit{subl} \\ \textsc{def} & : & \delta \end{bmatrix} \rangle$
∅
∅

There remains one question to be answered: what distinguishes a case from a construction that does not introduce a case? For example, does every preposition or postposition represent a case of its own? We think that this is not the case. We shall say that what distinguishes the genuine cases from other superficially similar constructions is that cases are selected by a lexical head and are therefore void of meaning. For example, there is no word to our knowledge selecting a PP with `entlang` (*along*). Therefore, there is no corresponding case. To say that a case is void of meaning is not to say that it is meaningless in all respects. For example, the fact that an NP carries nominative case distinguishes it from an NP that carries accusative case. Therefore, to have nominative rather than accusative case makes a difference. Yet, we claim that there is no inherent meaning in the nominative other than that, ie the only meaning is its distinctness from other cases. Now, in order to justify the claim that cases generally have no meaning we shall have to distinguish between free and selected case. This is the subject of the next section.

5.5 Selected and Free Case

At the end of the previous section we have already said that there are case endings that do not function as cases. Here we are going to look deeper into this phenomenon. To get the terminology clear, we shall makes a distinction between *free* and *selected case*. This will first of all establish a basis for the classification of cases, and secondly allow to substantiate the claim that cases are basically meaningless. To approach the problem, let us consider the noun phrase in Finnish. We have full concord within the noun phrase, so an adjective agrees with the modified noun in case. So, when we have a phrase like 'in the big shop', we shall find that 'big' agrees in case with 'shop'. When we apply our mechanism of case assignment to the noun phrase we shall find that both times the case suffix is added, it contributes to the meaning of the phrase by saying something like 'in the location of the object'. So, the resulting structure is appropriately paraphrased as

(5.91) in the location of x which is big and in the location of x which is a shop

The problem with this analysis is not so much that it is redundant but more that it can lead to inconsistent paraphrases as soon as a case ending has several basic meanings. Indeed, the inessive has other meanings as well, it is used for the time interval within which something happens, or the material from which something is made. So, in our paraphrase we would have to keep track of the kind of meaning that we have chosen to start with. Obviously, this is not a satisfactory analysis. Rather, it would be better if the case meaning would be expressed only once, namely when the phrase is complete. In all other instances the case suffix just signals agreement, nothing else. However, this leaves us still with two kinds of cases, those which just signal agreement and those which signal the 'true' meaning. To eliminate this, we shall assume that case does nothing than signal agreement. So, when an element is put into the allative in Finnish it means that it just has the allative case, nothing more. This makes matters particularly easy for such words which select complements bearing a certain case. Since the case is void of meaning, there is nothing of semantic value that intervenes between the word and its complement, there is only the syntactically required case ending. This is basically the story of selected case. Any case, when it is selected, is void of meaning.

If that is so, then what about cases that are not selected? We shall assume that there are empty prepositions which can turn the element in the required case into a meaningful element. (An alternative analysis, where this preposition is actually the case ending of the determiner or the head, is also viable, but not particularly more elegant than this one.) The flexibility in the meaning of the cases can be traced to the empty prepositions. For example, we assume that there is a phonologically empty preposition ILLA, which requires illative case and returns an adverbial. For example, the following example must be glossed using this preposition.

(5.92) `menen isoon huoneeseen.`
`menen`(ILLA(`isoon huoneeseen`))
I walk ILLA big-ILL room-ILL
I walk into a big room.

The preposition ILLA not only gives the complement its true meaning, it also removes its case and turns the expression into an adverbial.

(5.93)

/ILLA/
$\langle e : \Diamond⦶ : [\quad] \rangle$ $\langle x : \triangledown⊜ : [\textsc{case} : \mathit{ill}] \rangle$
$\varnothing$
$\mathsf{in}'(y, x, e)$

Here, $\mathsf{into}'(y, x, e)$ means that y moves into x during event time. Obviously, the identity of y must be established as well. One can say that y is often either the subject or the object of the verb, though the matter is sometimes quite complex. If these notions are encoded in the semantics they can be used here as well. We shall leave that matter unresolved here, noting only that the construal of adverbials is a difficult problem. The relevant properties that determine construal must be semantically encoded in order for the present proposal to work properly.

Now notice that after application of the preposition there is no case left. This is important. This distinguishes free case from selected case. The case has been exchanged for the meaning so to speak. Before we move on, we shall however refine our proposal somewhat. First of all note that local cases have two layers, which we have characterized as the modal and the configurational layer. We shall assume here that these layers are undone by a sequence of two prepositions, one defining the location and another defining the mode (if applicable).

(5.94) `menen isoon huoneeseen.`
`menen`(TO(IN(`isoon huoneeseen`)))
I walk TO IN big-ILL room-ILL
I walk into a big room

So, ILLA is now the sequence of IN and TO. To undo the illative case, these two prepositions are needed in sequence. The first is IN. Rather than defining an adverbial, we shall assume that it defines simply a location. To make that work, we need to introduce two things: a separate entity of type location (denoted by variables ℓ, ℓ') and an we need an explicit mode feature for locations whose values are *stat*, *cof* or *coin* (and possibly more in other languages) for Finnish.

(5.95)

/IN/
$\langle \ell : \vartriangle\ \varnothing : [\text{MODE} : cof]\rangle$ $\langle x : \triangledown\ominus : [\text{CASE} : iness]\rangle$
$\lambda t.\mathsf{in}'(\mathsf{loc}'(x)(t))$

Here, in' is a function from locations to locations. (Given the location of a box, it returns the interior of that box, for example.) We have added the time parameter to make the example more realistic. So, IN takes an object and returns a location at a parametric time point, and the case is reduced to a mode. Notice that the output is not exactly a location but a location changing in time. To remove the mode feature, the preposition TO is used.

(5.96)

/TO/
$\langle e : \Diamond\oplus : [\text{TYPE} : -state] :: [\text{PRED} : t]\rangle$ $\langle \ell : \triangledown\ominus : [\text{MODE} : cof]\rangle$
$\varnothing$
$\mathsf{move\text{-}to}'(\mu(e), \ell, t)$

Here, $\mu(e)$ is a function that takes an event and returns the element canonically moving in that event. $\mathsf{move\text{-}to}'$ takes an object x, a parameterized location ℓ and a time interval t and says that x is moving during t into ℓ. The Finnish locatives can therefore be defined by means of a combination of two types of elements as shown

Table 5.11: Decomposing the Finnish Locatives

inessive	STAY⌢IN
elative	TO⌢IN
illative	FROM⌢IN
adessive	STAY⌢AT
allative	TO⌢AT
ablative	FROM⌢AT

in Table 5.11. To see a different example, we shall discuss the locative cases of Tsez (see [19]). Cases in Tsez are similar to those of Avar. However, in between the configurational element and the modal element lies an optional element which tells us whether the location is visible (non–distal) or not (distal). For example, we have the case ending `-xo-r`, which contains the element `-x(o)` denoting the 'at' configuration, and the element `-r` for the lative mode. So, this is the allative case ending. If we want to express the fact that motion is towards a location which is distal, we choose `-x-āz-or`, where `-āz` is the suffix denoting non–visibility. Now, for the system of cases of Tsez, so we claim, the discrimination between distal and non–distal is irrelevant. We are still dealing with the same case, whether it is spelled `-xo-r` or `-x-āz-a-r`. Now, if Tsez was like Finnish, then we would be in trouble. For if there is an adverbial carrying the allative case, it would be built by first constructing a DP in the allative (using the morphological case) and the prefixing some empty prepositions corresponding to the configuration and the mode. This has for consequence that we cannot distinguish between distal and non–distal. [5] However, in Tsez, there is no NP–internal agreement, and therefore the cases can be treated as phrasal affixes, in which cases they are the full carriers of meaning in the adverbial case. Then the distinction between distal and non–distal is straightforward.

With these prepositions we can actually analyze the entire system of locative cases, just adding a few more (or removing some) if necessary. Namely, seen in this way the mode and the configuration are made independent parts and can

[5]The reader is made aware of the fact that this argument rests on the assumption that there is no selectional distinction between distal and non–distal cases. If there were, then of course we would have to assume a three–grade series of empty prepositions, one for the configuration, one for the distalness and the third for the mode.

therefore be combined independently. This makes sense also from another point of view. If I want to say where I am going, then what is selected by the verb with respect to the locative is only its mode and not its configuration. Or, to give a different example, the verb `saapua` ('to arrive') is construed with cofinal mode in Finnish and not with stative mode as in English. When we want to say where we arrived, then the actual case depends in Finnish on the city. The default for cities is to use the 'in' configuration (and hence illative with `saapua`), while some (Finnish) cities exceptionally require the 'at' configuration, and so the allative case with `saapua`:

(5.97) `Saapumme Lontooseen.`
arrive-1.PL London-ILL
We arrive in London.

(5.98) `Saapumme Tamperelle.`
arrive-1.PL Tampere-ALL
We arrive in Tampere.

If we would use just one preposition we could not state the regularity that Tampere is construed with the 'at' series and London with the 'in' series, while the mode is determined by the verb or another head. However, if we separate mode and configuration then we can say that for certain cities the canonical location is construed with the 'at' series (this needs to be marked in the lexicon), while all others are construed with the 'in' series. However, once the location is formed using the right configurational preposition, all irregularities disappear. The mode can for example freely be chosen by the verb. So, all that we need to do is to say that the verb `saapua` selects a location in the cofinal mode. The corresponding verb in Hungarian, `érkezik`, selects static mode, but the same idiosyncrasies can be noted with place names. And so on in other languages.

There are more advantages to this approach. We note for example that directional locatives cannot be used to modify nouns. Only static locatives can. (There is an exception with the phrase 'from' as in *Jesus from Nazareth*. This has coinitial mode.) Hence, we assume that a location can be turned into an adjective by yet another preposition LOC, whose lexical entry is as follows.

(5.99)
$$\begin{array}{c} /\textsc{loc}/ \\ \hline \langle x : \lozenge\oslash : [\textsc{prd} : -] :: \left[\begin{array}{lll}\textsc{prop} & : & p \\ \textsc{pred} & : & t\end{array}\right]\rangle \\ \langle \ell : \triangledown\oslash : [\quad]\rangle \\ \hline \varnothing \\ \hline q \doteq p \wedge (\forall i \in t)\mathsf{loc}'(x)(i) \subseteq \ell(i) \end{array}$$

Here, i is a variable over time points. So, when I say, for example, that this is a book on the table, I mean to say that this is a book which happens to be on the table at the particular time when I utter the sentence. There are a number of exceptions to this rule that we shall briefly discuss. With certain nominals denoting events (for example `trip`) all kinds of locatives are appropriate. This needs to be accounted for eg by allowing a mismatch in syntactic and semantic category.

Now, although the idea of an empty preposition strikes one as absurd in a language that actually has a case that has a locational meaning, it becomes much easier to acknowledge in languages where this is not so. We take for comparison German. Here, the corresponding local cases must be formed using overt prepositions. These can be seen now as the overt equivalents of the Finnish empty prepositions (or sequences thereof). The only difference between Finnish and German apart from this is the fact that Finnish has a more elaborate morphological case system. However, notice that even though we now have no problem with the existence of locative prepositions there is now the opposite fact to be acknowledged: some of these prepositions are actually void of meaning. The German preposition `auf` (with accusative), for example, means literally translated *onto*. However, when selected by a particular word (for example `sich beziehen` 'to refer') then it means—nothing. For in that case it is only a case marker, changing the case from accusative to sublative. If however the preposition has its natural meaning, so to speak, then it takes an accusative complement and returns an adverb, just as in the example in Finnish.

The present approach does not emphasize so much the border between cases and prepositions but between selectedness and freeness. The reasons for not distinguishing cases and prepositions have been discussed earlier. The distinction between being selected and being free is simply that of the semantics: free case carries meaning, selected case does not. This has implications also for the organisation of the NP in those languages that have NP internal case agreement. In these languagues the NP internal agreement is of course completely formal, and there is a need to undo the formal cases by means of (empty) prepositions to make their meaning come out. In languages that do not have NP internal case agreement, this is not necessary. We shall only have to assume that case markers are syntactically and semantically like prepositions, only that they have different morphological properties. This is the system of English and German, for example. Of course, since agreement carries only as far as Layer I, we shall find that many languages need a mixture of these two strategies.

Notes on this section. The distinction between free and selected case has not attracted much attention within grammatical theory. Most theories take cases to be selected or assigned in one or the other way. In [71], page 71, ft. 26 this problem is explicitly acknowledged. The proposal is to say that free cases create functions that are distinct from the others. This is needed to satisfy the functional uniqueness law. Within LFG, however, this has been accounted for by allowing adjuncts to come in sets (see [51]). Our solution is independent of the framework chosen. As long as one is prepared to assume some empty prepositions, it can be transferred to other frameworks as well. It has the advantage over the just mentioned LFG approach that the adjuncts are in no sense of the word selected by the head or predicate, so no constraint of the sort that the functional uniqueness law provides constrains the occurrence of the adverbs. Moreover, it makes no difference whether such adverbs are formed by means of a simple adverbial (for example `now`) or by a PP (`at this moment`). In our approach, adverbs take the verb as their argument, as expected.

5.6 Shells of Case

In a small number of languages, mainly spoken in Australia and the Caucasus, a single noun or adjective may appear with several case suffixes. This arises when a noun appears embedded in several constituents, each of which is assigned a different case. Here is a particularly complex example from Kayardild (see [30]).

(5.100)

`Maku-ntha`	`yulawu-jarra-ntha`	`yakuri-naa-ntha`
woman-OBL	catch-PAST-OBL	fish-MABL-OBL
`dangka-karra-nguni-naa-ntha`		`mijil-nguni-naa-ntha`
man-GEN-INST-MABL-OBL		net-INST-ABL-OBL

'The woman must have caught fish with the man's net.'

(Here, OBL stands for *oblique* and MABL for *modal ablative*.) The oblique signals some aspect of the sentence. Notice that the verb also carries a case marker for OBL. This may be taken as a sign that these are not really case suffixes, but this is a matter of theoretical decision. We follow here the linguistic tradition and call them cases. The complement of the verb is marked by the ablative, and additionally it shows case concord with the subject and the verb. The instrument however also takes the modal ablative as well as the oblique, and finally the possessor takes all these three suffixes and the genitive.

The phenomenon of several (case) suffixes is called *Suffixaufnahme* (lit. taking up of suffixes), but it is not clear that what it involves is taking up a suffix. We will however continue to refer to this phenomenon as Suffixaufnahme. Kayardild is perhaps the extreme case of Suffixaufnahme, and together with Martuthunira and Warlpiri certainly the best documented case within the group of Australian languages. Caucasian languages also exhibit Suffixaufnahme, but here the phenomenon is generally limited to possessive constructions. In Old Georgian there is double case marking in that there is independent adnominal case marking (genitive case) and adverbal case.

(5.101) `sarel-ita mam-isa-jta`
name-INST father-GEN-INST
with father's name

In the remainder of this chapter we shall develop a model for Suffixaufnahme, which will also deepen our understanding of the role of variables in referent systems. To begin, we shall agree to call the slots in which the cases appear **shells**. The word `dangka-karra-nguni-naa-ntha` has four case shells, which are numbered from left to right. The innermost shell is therefore the first shell. In the first shell we find the genitive case, in the second shell the instrumental, in the third the modal ablative, and in the fourth shell the oblique. So, cases not only come in layers, they come also in shells. We should also note that a case–shell may carry more than just a case suffix. Therefore, it is important to distinguish the case suffix from the shell in which it occurs. We will see that the shells are directly reflected in the semantics.

The case suffixes are ordered universally according to what is known as the **iconic order of suffixes**. (See [30] for an apparent exception to this ordering.) This order is the following. The most recently assigned case is the one we find closer to the root. Suppose that X assigns instrumental to its sister constituent, [Y Z]. And suppose that Y assigns genitive to Z. Then the suffix ordering we find on Z is Z-GEN-INST and not Z-INST-GEN, as we see in (5.101).

(5.102)
$$\begin{array}{ccccc} & \overset{\text{INST}}{\oslash} & & \overset{\text{GEN}}{\oslash} & \\ [\mathsf{X} & & [\mathsf{Y} & & \mathsf{Z}]] \end{array}$$

In order to account for the presence of shells in the semantics we shall refine our notion of a variable. In particular, we shall introduce a new mechanism of handling variable names. One may think of a particular case simply as an address into which a particular content can be stored. A verb with several complements can tell which complement is which simply by looking at the cases. We may alternatively think of the verb as looking into certain registers, which are named by cases, into which the noun phrases store some content. To do that, we shall say that a variable name is in effect a sequence of cases followed by a special symbol, here •, whose meaning becomes clear in a minute. So, here are names for variables:

(5.103)
$$\begin{array}{ll} \bullet, & \text{NOM}^\frown\bullet, \\ \text{ACC}^\frown\bullet, & \text{INST}^\frown\text{GEN}^\frown\text{ABL}^\frown\bullet \end{array}$$

Since the number of cases that can be stacked in this way in unbounded, we actually have an infinite resource of names. These names are now used instead of the variable names e, e', x, y and so on. The lexical entry for a nonrelational noun now looks like this:

(5.104)

/dog/
$\langle \bullet, \vartriangle\ \varnothing, v\rangle$
$\varnothing$
dog′(•)

The case of a relational noun is already interesting.

$$(5.105)\quad \begin{array}{|l|}\multicolumn{1}{c}{\texttt{/teacher/}}\\ \hline \langle\bullet,\ \vartriangle\ \varnothing,\nu\rangle;\\ \langle\textsc{gen}{}^\frown\bullet,\triangledown\obar,\nu'\rangle\\ \hline \varnothing\\ \hline \mathsf{teach}'(\bullet,\textsc{gen}{}^\frown\bullet)\\ \hline\end{array}$$

The complement noun is now simply identified by the fact that its variable is $\textsc{gen}{}^\frown\bullet$, which will imply that has genitive case. In order to organize the case system we shall still keep the case feature in the name space, but in contrast to the case feature, the case element of the case sequence is iterable. This allows to organize the layers of case together with the shells.

The main idea is now that the actual case suffixes of the language act as substitution devices. When merged with a particular word, they substitute $\bullet$ throughout the structure by a sequence. For example, if we apply instrumental case to the word dog, we get

$$(5.106)\quad \begin{array}{|l|}\multicolumn{1}{c}{\texttt{/dog}{}^\frown\textsc{inst}\texttt{/}}\\ \hline \langle\textsc{inst}{}^\frown\bullet,\ \vartriangle\ \varnothing,\nu\rangle\\ \hline \varnothing\\ \hline \mathsf{dog}'(\textsc{inst}{}^\frown\bullet)\\ \hline\end{array}$$

Similarly, if accusative is attached to `teacher` we get

$$(5.107)\quad \begin{array}{|l|}\multicolumn{1}{c}{\texttt{/teacher}{}^\frown\textsc{acc}\texttt{/}}\\ \hline \langle\textsc{acc}{}^\frown\bullet,\ \vartriangle\ \varnothing,\nu\rangle;\\ \langle\textsc{gen}{}^\frown\textsc{acc}{}^\frown\bullet,\triangledown\obar,\nu'\rangle\\ \hline \varnothing\\ \hline \mathsf{teach}'(\textsc{acc}{}^\frown\bullet,\textsc{gen}{}^\frown\textsc{gen}{}^\frown\bullet)\\ \hline\end{array}$$

We can already see that through this procedure the variable names get longer and longer. There is no danger that different variables will suddenly end up being the same, since the substitution is injective.

The formal proposal that we are going to make follows exactly this idea. Notice that when we have a transformer it was always so that the transformer can change the name of the variable, though the name was something different from the symbol that was written down to refer to this variable (eg x or y), which we may call the **register**. The register was previously manipulated only by the merge operation. Now we shall assume that merge is actually the Zeevat–merge, so that it does not even manipulate the register. If we want to change the register, we have to do that explicitly, by means of cases. Conceptually, we now think of the register as part of the name, and this gives us the chance to let trnsformers do the job of changing the register. So, the argument structure of the nominative suffix is as follows:

$$\begin{array}{c}/\textsc{nom}/\\ \begin{array}{|l|}\hline \langle\bullet \mapsto \textsc{nom}^\frown\bullet, \Diamond⧀, \nu\rangle\\ \hline \varnothing\\ \hline \varnothing\\ \hline\end{array}\end{array} \tag{5.108}$$

By definition of the symbols, this element looks for • to its left and substitutes it by NOM⌢•. For example, we have

$$\begin{array}{c}/\texttt{teacher}/\\ \begin{array}{|l|}\hline \langle\bullet,\ \vartriangle\ \varnothing, \nu\rangle;\\ \langle\textsc{gen}^\frown\bullet, \triangledown⦶, \nu'\rangle\\ \hline \varnothing\\ \hline \mathsf{teach}'(\bullet, \textsc{gen}^\frown\bullet)\\ \hline\end{array}\end{array} \bullet \begin{array}{c}/\textsc{nom}/\\ \begin{array}{|l|}\hline \langle\bullet \mapsto \textsc{nom}^\frown\bullet, \Diamond⧀, \nu\rangle\\ \hline \varnothing\\ \hline \varnothing\\ \hline\end{array}\end{array}$$

$$= \begin{array}{c}/\texttt{teacher}^\frown\textsc{nom}/\\ \begin{array}{|l|}\hline \langle\textsc{nom}^\frown\bullet,\ \vartriangle\ \varnothing, \nu\rangle;\\ \langle\textsc{gen}^\frown\textsc{nom}^\frown\bullet, \triangledown⦶, \nu'\rangle\\ \hline \varnothing\\ \hline \mathsf{teach}'(\textsc{nom}^\frown\bullet, \textsc{gen}^\frown\textsc{nom}^\frown\bullet)\\ \hline\end{array}\end{array} \tag{5.109}$$

We should stress that what happens is that when two structures are merged, • and • get identified. This is why whe call this the Zeevat–merge. However, since the second structure is a transformer, the result is defined by the Zeevat merge followed by the substitution of NOM⌢• for •. This is what means • ↦ NOM⌢• by our conventions. Hence by this move the handling of variable names is made completely explicit. There are no hidden substitutions.

The proposal predicts the iconicity of case marking in the following way. The case name is also iconic if read from left to right, or, which is somewhat neutral, it is anti–iconic if seen from the top, which is represented by •. The variable associated with Z-GEN-INST is GEN⌢INST⌢• and not INST⌢GEN⌢•. So, the register gives the sequence of case–suffixes counting from bottom to top. On the other hand, if `Z` assigns accusative case to some complement `W`, then the register of `W` in the representation of `Z` is $\textsc{acc}^\frown\bullet$. If we look at the register in `Z`-$\textsc{gen}$-$\textsc{inst}$, we find that `W` now has the register $\textsc{acc}^\frown\textsc{gen}^\frown\textsc{inst}^\frown\bullet$. So, the corresponding cases are ordered as follows: `W`-$\textsc{acc}$-$\textsc{gen}$-$\textsc{inst}$. Hence, iconicity is preserved.

This mechanism is complelety independent of whether cases are realized by suffixes, prefixes, clitics or appositions. Moreover, it can be used even when there is no case stacking is involved, as in most languages of the world. We shall discuss this point further below. Notice that we have allowed ourselves great freedom in the manipulation of registers if we allow transformers to do explicit substitutions. The question is therefore whether this proposal is motivated independently. This is the case. In the discussion of the layers of case we have concluded that the outer layers actually transform the case name. For example, the German preposition an when selecting the accusative transforms the case name of its complement from accusative to allative. Hence, we shall analyze this preposition as follows:

$$(5.110)\qquad \begin{array}{c} /\mathsf{an}/ \\ \begin{array}{|l|}\hline \langle \textsc{acc}^\frown\bullet \mapsto \textsc{all}^\frown\bullet, \Diamond\olessthan, \nu\rangle \\ \hline \varnothing \\ \hline \varnothing \\ \hline \end{array} \end{array}$$

Notice that the transformer defines a substitution that is not injective anymore. But the cases of noninjectivity are narginal. We do not expect however that there are transformers that cut off a case name, eg $\textsc{nom}^\frown\bullet \mapsto \bullet$. This would make the substitution not injective and lead to dangerous clashes in the registers.

Cases may at the same time transform the register and have content. For example, Martuthunira has a privative case. The case suffix would have the following semantics.

$$(5.111)\qquad \begin{array}{c} /\textsc{priv}/ \\ \begin{array}{|l|}\hline \langle \bullet \mapsto \textsc{priv}^\frown\bullet, \Diamond\olessthan, \nu\rangle \\ \hline \varnothing \\ \hline \neg \begin{array}{|l|}\hline \textsc{priv}^\frown\bullet \\ \hline \mathsf{have}'(\bullet, \textsc{priv}^\frown\bullet). \\ \hline \end{array} \\ \hline \end{array} \end{array}$$

So, if it is attached to some head x, then it denotes the property of being without x. We remind the reader of the fact that we have presented a different analysis before. We have argued that for example the illative, when it is used adverbially, is actually 'killed off' by an empty preposition. This suggests that semantic cases do not induce a substitution in the register. Instead, the relevant variable is simply removed. For example, the preposition ILLA gets the following translation.

(5.112)

/ILLA/
$\langle \bullet, \Diamond\obar, [\quad] \rangle$,
$\langle \text{ILL}^\frown\bullet, \triangledown\ominus, \nu \rangle$
$\varnothing$
move-into′($\bullet$, ACC$^\frown\bullet$, ILL$^\frown$)

Notice that the present proposal, if correct, requires extra machinery for the identification of variables such as the mover in the case of ILLA. The semantics specifies that the mover moves to the object denoted by ILL$^\frown\bullet$ during the time of the event $\bullet$. But how do we address the object? Its register is a sequence of cases.

We have spoken above about the fact that certain morphemes may come in between two case markers. Here is a longer stretch of text in Martuthunira, which exemplifies once more the case stacking feature of this language.

(5.113)

A:Ngayu	kangku – lha	mayiili – marnu – ngu	kulhampa – arta.
I	take-PAST	SoSo+POSS:1-GRP-ACC	fish-ALL

I took a group of my grandchildren for fish.

(5.114)

B:nganangu – ngara	pawulu – ngara?
whoGEN-PL	child-PL?

Whose children are they?

(5.115)

A:Ngurnu – ngara – a	yaan – wirriwa – wura – a.
thatOBL-PL-ACC	spouse-PRIV-BELONG-ACC

(I took) the ones who belong to the one who is without a spouse.

(5.116)

B:Ngaa,	purrkuru	pala.	Ngarraya – ngu – ngara – a.
Yes	Okay	IT	niece-GEN-PL-ACC

Yes. Okay that's it (I understand). (You took) niece's ones.

What we see is not only the fact that there are various case markers, and that they can be stacked. But that there are for example elements expressing plurality and that they may be positioned before or in between the case markers. The difference is easily explained if we look at the semantics. We propose the following semantics for plurality:

(5.117)

/PL/
$\langle \bullet, \blacklozenge \otimes, [\text{NUM} : \star \mapsto pl] :: p \rangle$
$\varnothing$
$\sharp \bullet > 1.$

This has the following consequence. If plural is inside the case ending, it will attribute plurality to the object in question, but not if it is outside. This is so since the case marker if applied first will stack the variable away. Suppose namely we attach accusative to the word `teacher`. Then the noun register is $\text{ACC}^\frown\bullet$. If we next attach plural, then we get the information that $\bullet$ is a group of cardinality > 1, and not that $\text{ACC}^\frown\bullet$ is a group of cardinality > 1. The result is different when we first attach plural and then accusative, for then we get the result that $\text{ACC}^\frown\bullet$ is a group of teachers. A case in point is (5.116), in particular the word `ngarraya-ngu-ngara-a`, glossed as niece-GEN-PL-ACC. Here, if the plural marker would be inside the genitive, then it would means something like 'the one belonging to the nieces'. So we immediately get the result that number is inside case, without any stipulation. The same for gender and class and person. We shall return to this phenomenon in the next section.

5.7 Parameters of Case Marking

We shall close this chapter with some considerations concerning the differences in case marking. The are various parameters along which case marking patterns can be differentiated. One is morphological: cases can be affixes, clitics or separate words. (There may in additional be fusional case, but we shall not pursue that further.) Second, we may parametrize the direction from which cases are attached. And third, we may classify case endings as to whether they attach to the word individually or to the phrase. In case of cases which are affixes, the distinction that we mean is that between a word affix and phrasal affix. This means informally that the affix must be repeated at each individual word of the phrase while a phrasal affix appears only once per phrase. One note of caution is in order. We have several layers of case, and the cases of Layer I are characterized as the morphological cases. Since there are plenty of languages that have cases of Layer I and II, it is of no use to speak of languages marking case exclusively by means of affixes or exclusively by means of appositions. It would be worth investigating whether cases that are clitics are to be counted as Layer I or as Layer II. Here we shall assume that they are of Layer II. Hence the classification must distinguish the cases of different layer. Moreover, the distinction between word level and phrasal does apply only to affixes. That is to say, I know of no language in which cases of Layer II or higher are word level. Lets take as our first example German. German has Layer I cases, and they are suffixes. These suffixes are word level, and so must be added at each word separately. By contrast, Hungarian has Layer I cases that are phrasal suffixes. Now, cases of Layer II are generally prefixing, and cases of Layer III may be either prefixing or suffixing. This shows that there is a mixture in the directionality of cases.

In what is to follow we shall be concerned therefore only with cases of Layer I, since these cases present the most elaborate system of distinctions. Turkish has five cases of Layer I, and they are phrasal affixes. The following illustrates this. (See [58].)

	[ben	-im	hasta	ol	-duğ	-um]	-u
	I	-GEN	sick	be	FNOM	1.SG	ACC
(5.118)	bil	-iyor	-mu	-sun?			
	know	-PRF.PROG	-QSTN	-2.SG			
	Did you know that I was sick?						

Here, the accusative case ending is attached to the entire clause, which has been nominalized. In languages which have phrasal affixes, these may actually end up in a sequence, giving the appearance of suffixaufnahme. The following is found in Sumerian (see [79]).

(5.119) é lugal-ak
house king-GEN
house of the king

(5.120) é lugal-ak-a
house king-GEN-LOC
in the house of the king

(5.121) é šešlugal-ak-ak-a
house brother king-GEN-GEN-LOC
in the house of the brother of the king

Here, the case is a phrasal suffix, and the genitive complement also follows the head noun. Hence the last example is to be bracketed as follows.

(5.122) (é (šeš (lugal)-ak)-ak)-a

This is admittedly a rare example, though we find in the abovementioned source also examples from Late Elamite and Kanuri (Nilo–Saharan). Languages usually avoid such complications by choosing for the complement just the other way. In English, for example, `of` chooses its complement to its right (so is prefixing), but it also puts itself to the right (in other words it is suffixed). Genitive `'s` on the other hand, is a suffix, and it puts itself in the front. Compare (5.123) and (5.124) with how it would be like if `of` and `in` were suffixing (5.125) or `'s` was a prefix (5.126):

(5.123) `in the house` [`of the brother` [`of the king`]]

(5.124) `in` [[`the king`]`s brother`]`'s house`

(5.125) `the house` [`the brother` [`the king of`] `of`] `in`

(5.126) `in s'`[`s'`[`the king`] `brother`] `house`

The latter patterns sound extremely clumsy, but (5.125) is exactly what we find in Sumerian. Nevertheless, one cannot say that such constructions are disallowed in general. Let us take some German examples. We can stack prepositions, even though stacking more than two is strongly disfavoured:

(5.127) `in [in Buchläden gekauften] Büchern`
in books bought in bookstores

(5.128) ??`in` [`in` [`in Berlin ansässigen`] `Buchläden gekauften`] `Büchern`

There are alternative constructions that avoid such stackings. Of course, with cases of Layer I such stacking can never occur, by the morphology of German.

We conclude that the presence of stacked cases does not present evidence of suffixaufnahme, since we must always reckon with the fact that the case marker is only a phrasal affix. However, in Georgian and Kayardild it clearly is not. It is a word level affix.

Let us therefore return to cases as word level affixes. Here is a rather spectacular example of cases appearing in up to three shells, each carrying extra material.

This example is from Hurrian texts of Boğazköy, quoted from [98].

(5.129) unni=ma ^{D}Teššub=va šarri=ne=va evre=n(i)=ne=na ...
en(i)=na=až=(v)a egli=ve=NE=ve=NA=až=(v)a
šubri=ve=NE=ve=NA=až=(v)a ... un=a
now=PRT Teššub-DAT king-ART.SG-DAT lord-INDIV-ART.SG-DAT ...
god-ART.PL-PL-DAT salvation-GEN-CARR:SG-GEN-CARR:PL-PL-DAT
šubri – GEN – CARR.SG – GEN – CARR:PL – PL – DAT...*come* – INTRNS
Now he comes to Teššub, to the king, to the lord ..., to the gods of the saviour (lit. of the one of salvation), of the one of the šubri ...

Here is a division of a word into shells:

	šubri	=ve	=NE=ve	=NA=až=(v)a
(5.130)	*šubri*	-GEN	-CARR:SG-GEN	-CARR:PL-PL-DAT
	root	1	2	3

We find three shells, each ended by a case suffix, containing some plurality markers and some extra suffixes. Additional evidence that the plural marker must be put into the same shell with the following case is given by

	en(i)=n(a)=až=už	attani=ve=n(a)=až=už
(5.131)	god-RLT:PL-PL-ERG	father-GEN-RLT:PL-PL-ERG
	the gods of the father	

Here we see that the plural goes with the ergative, not with the genitive. [6]

In contrast to our earlier examples from Martuthunira, we do not only find plural expressed once, but several times over in the same word, but it does seem that the plural in different shells fulfills the same function as the case suffixes themselves: they get copied from higher heads. Thus, their presence is mandatory. We can model this by assuming that registers are not only sequences of case namers, but sequences of AV–structures containing a case name and a number name. So, the word analysed above has the following register:

(5.132) $$\begin{bmatrix} \textsc{case} : gen \\ \textsc{num} : \star \end{bmatrix}^\frown \begin{bmatrix} \textsc{case} : gen \\ \textsc{num} : sg \end{bmatrix}^\frown \begin{bmatrix} \textsc{case} : dat \\ \textsc{num} : pl \end{bmatrix}^\frown \bullet$$

[6]The reason for a different way of glossing Hurrian is that the example is taken from a different source, namely [101].

So, not only a case name is specified but also a number, and this is done in each element of the sequence. However, we see also that sometimes number is not specified. This means that there is room for exceptions. For Martuthunira, for example, it seems right to assume that number marking for the intermediaries is not obligatory. (However, note that singular is not expressed in Martuthunira.) However, if number is not marked, the two registers are not identical. Hence we shall not speak of identity of registers but of **unifiability**. Two registers $\langle \alpha_i : i < n \rangle$ and $\langle \beta_i : i < n \rangle$ are unifiable if for each $i < n$: α_i and β_i are unifiable. Two elements are unifiable in Martuthunira if they carry the same case name, and they are unifiable in Hurrian if they carry the same case and number feature. This generates the complex sequences of case and number, if number is inside case. This seems to be the case. Note however that Hurrian plural marker `-až` can follow the possessive suffix. Therefore, `en(i)=iff=až=uz` (god-POSS:1-PL-ERG) can mean either 'my gods', 'our god' and even 'our gods' (sic!). [7]

Notes on this section. Case stacking often does not appear in its pure form. Dench and Evans [26] note a number of exceptions to the norm. There is for example a general rule banning the occurrence of two identical morphemes next to each other. This can lead to one of them being lost or one of them being replaced by the other or the whole sequence being replaced by a portmanteau morph. These complications must however be dealt with by the morphology.

[7] This is taken from Wilhelm [101]. There are a number of apparent inconsistencies such as the translation 'our gods' which would seem to require two plural suffixes, another being that `iff` is glossed as 1SgPoss, which in our notation is POSS:1.SG. But this flatly contradicts the translation 'our god(s)'.

Chapter 6

The Fine Structure of Names

Names must have fine structure. For as it turns out, we cannot simply throw away all of the name when we just want to get rid of the case. This means that we shall divide the entire name space into strata. In each stratum the names act individually. They can be manipulated independently of the names at the other strata. This will allow to treat gender and number agreement in anaphora. Further, we shall show in detail how the verbal morphology acts on the stratified name space. At the end we shall also discuss some phenomena of disagreement.

6.1 Stratifying the Name Space

Up to now we have discussed the mechanics of names as if names were a unit. This has proved to be successful for quite a number of applications. In this chapter and the next we shall actually shows that for a number of reasons this is inadequate. The proposal that we shall advance is that the name space is actually divided into several strata which act more or less independently. This will allow for parts of the name to get lost or instantiated rather than the entire name. This is needed for a number of reasons, which we shall discuss below. However, we shall start with an outline of stratified names spaces. Consider a verb looking for a subject. The argument structure of the verb is such that if it combines with the subject, the subject variable looses its name completely. Conversely, assume that the verb assigns to some actant a θ–role but no case, or a case but no θ–role. There is no way to represent this state of affairs in the present calculus. For if there is an argument handling statement for a variable, say x, that contains a name for x then the name may be underdefined, but there are no undefined parts of the name. In other words, we cannot have the following argument handling statement:

$$\langle x : \triangledown\ominus : \begin{bmatrix} \text{CASE} & : & nom \\ \theta\text{–ROLE} & : & \checkmark \end{bmatrix} \rangle$$

So, it may not be that the θ–role is unassigned while case is assigned. However, we have quite often made use of $\star$, which served as the generic undefined value. Hence the following was quite legitimate:

$$\langle x : \triangledown\ominus : \begin{bmatrix} \text{CASE} & : & nom \\ \theta\text{–ROLE} & : & \star \end{bmatrix} \rangle$$

Note that $\star$ is a genuine value. For example, we have allowed that Hungarian adjectives carry the feature [PL : $\star$]. Moreover, $\star$ is identified under merge, while a name that is lacking is not identified. For reasons that will hopefully become clear we shall argue that we do want to be allowed to leave parts of the name undefined. However, that shall not be any part of the name, and therefore $\star$ is not rendered superfluous.

A particular case where we want to separate two parts of a name is the φ–features of a noun on the one hand and the case feature on the other. There is plenty of evidence that the φ–features of a noun are alive even after the case feature has been lost. A case in point is pronominal reference. We can use a pronoun to refer back to an entity in the previous sentence. However, if we want to pick up a referent by means of its φ–features then they should be present in the representation. The φ–features are therefore independent of the case. A case in point is the following text.

(6.1) `Harry`^{1} `was talking to Susan`^{2}. `He`$_{1/*2}$ `was quite upset.`

(6.2) `Harry`^{1} `was talking to Susan`^{2}. `She`$_{*1/2}$ `was quite upset.`

The difference between (6.1) and (6.2) is only the gender of the pronoun. Yet it clearly determines which of the discourse referents is being picked up.

To account for such facts we shall assume that the name space is divided into several **strata**. The first stratum consists purely of the cases and another of the

φ–features. At each stratum a referent can either have a name or fail to have a name. So, we shall allow for referents to have just case and no φ–features (not even undefined φ–features) and we allow for a referent to have φ–features but no case at all. The latter type occurs when a verb has consumed an argument; then the referent looses its case but not its φ–features. Thus, in addition to allowing a name to be partially absent, we also allow operations to delete a name only partially. If that is so, we shall actually assume that certain diacritics must be independently specifiable at the different strata. The vertical diacritic, for example, may function independently. We shall generously extend this to the entire argument handling statement. This leads to the proposal of having different strata, at which a referent can either have an AIS associated with it or not. A nominal argument of a verb therefore shall have the following AIS:

$$(6.3)\quad \begin{array}{l}\langle x : \ \triangledown \oslash : \begin{bmatrix}\textsc{case} & : & nom\end{bmatrix}\rangle \qquad \leftarrow \text{Case Stratum} \\ \hline \langle x : \ \Diamond \oslash : \begin{bmatrix}\textsc{class} & : & fem \\ \textsc{num} & : & pl \\ \textsc{pers} & : & 3\end{bmatrix}\rangle \qquad \leftarrow \varphi\text{–Stratum}\end{array}$$

Here, the line separates the two strata. Therefore, the variable x has a name at both strata. Moreover, it has different diacritics at the different strata. For example, its case diacritic is $\triangledown$, which means that the case name disappears after merge. The φ–diacritic is $\Diamond$, and this will guarantee that the φ–features survive after merge.

If this proposal is accepted, a number of questions appear.

1. Is x identified when it is identified at least on one stratum, or shall we require it to be identified at both strata?

2. How many and which referents can be identified at the individual strata?

3. How are the strata organized?

The answers to these questions are nontrivial, and largely depend on the language in question. We shall discuss the last two questions later. The questions bear direct relevance to the definition of merge. We shall in fact not give exhaustive answers to all these questions, and so there will be some degree of freedom in setting up the correct definitions. This is mainly due to the fact that we are not confident enough to constrain the system so that it fits the data well. It seems also that languages exercise a certain degree of freedom with respect to the interaction of strata. But the specification will hopefully be clear enough. So, let us take two argument structures. A referent is identified if it is identified at one stratum. Hence, it is possible to identify a referent at the case stratum and not at the φ–stratum and conversely. So, a pronoun can pick up a referent by means of its φ–features alone. Hovwer, when a referent is shared at one stratum, then if it has a name at another stratum, it must be identified at that stratum as well. So, when say a referent is identified by its case name, then the φ–features cannot disagree in the two structures. Therefore, in the following merge, x is not shared:

(6.4)
$$\begin{array}{|l|}\hline \langle x : \nabla\ominus : \left[\textsc{case} : \textit{nom}\right]\rangle \\ \hline \langle x : \Diamond\ominus : \left[\begin{array}{lcl}\textsc{class} & : & \textit{fem} \\ \textsc{num} & : & \textit{pl} \\ \textsc{pers} & : & \textit{3}\end{array}\right]\rangle \\ \hline \end{array} \bullet \begin{array}{|l|}\hline \langle x : \triangle\ \varnothing : \left[\textsc{case} : \textit{nom}\right]\rangle \\ \hline \langle x : \Diamond\ominus : \left[\begin{array}{lcl}\textsc{class} & : & \textit{masc} \\ \textsc{num} & : & \textit{sg} \\ \textsc{pers} & : & \textit{3}\end{array}\right]\rangle \\ \hline \end{array}$$

The reason why x is not shared that although it has the same case name in both structures, its φ–name is different in them. Now, if x is the only referent the merge would fail. In other case it may well succeed, but x is simply not shared.

With regard to the first, notice the following. If the first option is taken, then word order is less free than if the second option is taken. Another question is whether one stratum should be taken as superior to the other. For example, we may say that if a referent is identified at the case stratum then it is identified anyhow; only when it is not identified at the case stratum then it may be identified at the φ–stratum if possible. The latter option is different from the previous ones. For consider the merge above. Rather than x not being shared, the last option will lead to a failure. x is shared at the case stratum, and therefore must be shared. But it cannot be shared at the φ–stratum, and therefore the structure clashes. The differences between these options may not always be apparent, since the pathologies that can distinguish between them may not arise. However, they agree in the following canonical situations:

1. If a referent has a name at both strata in both structures, then the names must agree in both strata.

2. If a referent has a name in the case stratum but not the φ–stratum in both structures, it is shared if it is shared at the case stratum.

3. If a referent has a name in the φ–stratum but not the case stratum in both structures, it is shared if it is shared at the φ–stratum.

(Note that we speak of a single referent being shared, rather than two referents being identified, which would be more accurate.) Now, there is an interesting intermediate case, namely when a referent has a name at one stratum in one structure but not in the other. For example, it is shared at the case stratum but it does not have a φ–name in one of the structures:

$$(6.5)\quad \begin{array}{|l|}\hline \langle x : \nabla\circleddash : \left[\textsc{case} : \mathit{nom}\right]\rangle \\ \hline \langle x : \Diamond\circleddash : \begin{bmatrix}\textsc{class} & : & \mathit{fem} \\ \textsc{num} & : & \mathit{pl} \\ \textsc{pers} & : & \mathit{3}\end{bmatrix}\rangle \\ \hline \end{array} \bullet \begin{array}{|l|}\hline \langle x : \triangle\ \varnothing : \left[\textsc{case} : \mathit{nom}\right]\rangle \\ \hline \varnothing \\ \hline \end{array}$$

In this case we allow x to be shared, and that it will inherit its φ–name from the left hand structure:

$$(6.6)\quad \begin{array}{|l|}\hline \varnothing \\ \hline \langle x : \Diamond\circleddash : \begin{bmatrix}\textsc{class} & : & \mathit{fem} \\ \textsc{num} & : & \mathit{pl} \\ \textsc{pers} & : & \mathit{3}\end{bmatrix}\rangle \\ \hline \end{array}$$

This will be a frequently encountered situation. For example, it allows morphological elements to add a certain name to a referent.

In this chapter we shall be mainly concerned with the verb and its internal structure. While we have previously concentrated on the agreement system, we shall now take a look at the organisation of the argument selection proper. It has argued at many places that it is not enough to consider verbs as taking arguments having a certain case; in addition to cases, we must consider also grammatical functions (subject, object) and θ–roles. We have previously added the grammatical functions into the name space on a par with the case. However, there are verbs which have subjects but assign no case to them (infinitives are argued to have this property), and verbs may assign case without a grammatical role (certain oblique arguments for example). Similarly, θ–roles are independent both of grammatical functions and case assignment. A verb may have a subject without a θ–role, and it may assign θ–roles without case. [1] If that is so, we cannot assume that the name space has one stratum. Rather, we shall assume that in addition to having a case

[1]Since we have a wider notion of case, this may actually not be apparent. Under a restrictive notion of case there are plenty of arguments that have θ–roles but no case. But with other selected arguments analyzed as having a case (in the wider sense), most of these examples will not do here. But infinitives are once again a case in point.

stratum and a φ–stratum we have a stratum for the grammatical functions (called **GR–stratum**) and a stratum for the θ–roles (called θ–**stratum**). Argument may be specified at each of the strata independently. Moreover, we shall assume that any argument with semantic content is actually has a θ–role, while some arguments only have a θ–role without bearing a grammatical function or a case.

We have studied case systems quite extensively earlier, so we shall confine ourselves with an outline of the remaining strata. The θ–stratum allows for AISs of the following form

1. $\langle x : \triangledown : [\vartheta : \mathit{ben}]\rangle$
2. $\langle x : \vartriangle : [\vartheta : \mathit{ben}]\rangle$
3. $\langle x : \Diamond : [\vartheta : \mathit{ben} \mapsto \mathit{thm}]\rangle$

There are no directional diacritics. θ–roles can be imported, exported or changed, although the latter is quite rare. The θ–role names correspond to the θ–roles that we have used so far. The GR–stratum is quite similar. However, the names there are even simpler. There exist only four kinds of grammatical functions: 1, 2, and 3 (or subject, object and indirect object) and P (predicate). Correspondingly, we shall have the following names:

1. $\langle x : \triangledown : [\textsc{gr} : \mathit{2}]\rangle$
2. $\langle x : \vartriangle : [\textsc{gr} : \mathit{2}]\rangle$
3. $\langle x : \Diamond : [\textsc{gr} : \mathit{2} \mapsto \mathit{3}]\rangle$

Writing down the argument structure of a verb can now become rather difficult, and so we shall confine ourselves to the minimum of notation possible.

Our theory of verbal structure is roughly as follows. A verb comes out of the lexicon with a certain argument structure. Various lexical processes can now change this argument structure. These lexical processes operate in three cycles. The first cycle arranges the θ–roles. It will add (or remove or change) certain θ–roles in the verbal argument structure. A typical example is the addition of a beneficiary. The second cycle consists in rearranging the grammatical functions.

A typical example is passivization. The third cycle (re-)arranges the case assignment. In the last cycle, verbal agreement is added. These cycles have been studied rather extensively in Relational Grammar (see [75] and [76]). Our adaptation of RG is different from the original conception in a number of ways, as we shall discuss.

Let us now return to the two questions raised above. The last question was how the strata are organized. We shall in fact assume that the structure of the strata is not uniform. While at the case stratum the AISs form a list, with access being regimented, this will not hold for the θ–stratum and the GR–stratum. Here, we assume these strata to be sets of AISs. The φ–stratum is a multiset or perhaps also a list. Now when we say that the GR–stratum and the θ–stratum is a set, this needs to be clarified. We have two basic scenarios in mind. The first scenario is that θ– and GR–strata allow only one referent to have a particular name, independent of any other stratum. The second scenario is that when x and y have no name both at the case stratum and the φ–stratum, then they must either have a different θ–name or a different GR, otherwise they are simply identical, that is, $x = y$. If however they have a case name, or at least one of them has a case name, then they can be distinct. This is responsible for a number of facts that are well attested across languages. Namely, there can be at most one subject, at most one object and most one indirect object, and similarly each θ–role can be assigned only once. Yet, case can be assigned to different arguments (for example accusative). So, while assignment of the same case to different arguments is possible, assignment of the same GR or the same θ–role to different arguments is not possible. However, these facts do not follow from the definitions above under the second scenario, and auxiliary assumptions would have to made. We shall therefore assume that the first scenario is correct.

As a last remark we shall assume that the conditions on merge apply as before to the case stratum: there must be exactly one referent that is shared, no more and no less. On the other strata there can but need not be an identification of other referents.

6.2 An Outline of Relational Grammar

Relational Grammar assumes that a sentence is organized using grammatical relations. A predicate can take certain arguments, and these arguments can be distinguished by the relation they bear with that predicate. There are many relations, but the most important ones from the standpoint of syntax and morphology are 1, 2 and 3. (See the introduction to Relational Grammar [77].) They correspond roughly to the more traditional terms of subject, direct object and indirect object. Consider the following sentence.

(6.7) `The dog bites the cat.`

Here, `bites` is the predicate, `the dog` bears the 1–relation and `the cat` the 2–relation with this predicate. As sentences can also assume relations with predicates, this schema is recursive.

A very important facet of RG is the fact that relations can be changed. A typical instance is the passive. The passive morphology on the predicate has as its effect that the constituent previously bearing the 2–relation with that predicate now bears the 1–relation.

(6.8) `The cat is bitten by the dog.`

We say that 2 is *advanced* to 1. One would therefore expect that there are now two constituents bearing the 1–relation with the predicate. This, however, is strictly forbidden. The law that forbids this is called the

> Stratal Uniqueness Law. In each stratum, for a given predicate there can be at most one constituent bearing a particular relation to that predicate.

What happens therefore with the previous subject? In RG it is said that it looses its grammatical relation, it becomes a *chômeur*. So, in (6.8), `is bitten` is the predicate, `the cat` bears the 1–relation with the predicate, and `by the dog` is a chômeur. To be a chômeur means in effect that one is not eligible for any syntactic operation based on relations. Chômeurs are frozen, so to speak. In distinction to the received notation we shall write $-$ to signal that a constituent is en chômage. This means that the name is lost. We also hold that this is not a relational sign.

This is in a nutshell the basic proposal of RG. There are of course many more operations on relations, and many more laws, and we shall encounter some of them as we go along. However, what we have just seen is enough to explain the basic tenets of RG. First, RG distinguishes two levels in (6.8): the first level, before passive morphology has applied, which is identical to the level associated with (6.7), and another level, after passive morphology has applied. These levels are called *strata*. The first is called the *initial stratum* the second the *final stratum*. There can be more than two strata; the non–initial and non–final strata are called *intermediate*. The syntactic representation for (6.8) contains both strata, not just one. This is important. For there are syntactic processes which are sensitive for the relations as they are in the initial stratum and other syntactic processes which are sensitive to the relations as they are in the final stratum. For example, reflexives must be bound by an antecedent which bears a higher relation; however, the comparison is made in the initial stratum and not in the final stratum. The relations are ordered as follows:

(6.9) $3 < 2 < 1$

So, in Russian (as in many other languages) a reflexive must be bound by some nominal whose relation is higher in the initial stratum. This is why in passives a reflexive can occupy the subject position. If one moves higher in the hierarchy one is said to be **advanced**, and if one moves lower one is said to be **demoted** or to **retreat**. It is possible also to be raised out of an embedded sentence (this is called **ascension**) but we will not be concerned with this possibility. Passive is nothing but 2–to–1 advancement. We shall note here that the role changing operations have one thing in common, namely that only one constituent changes its role. Otherwise, we could define another variant of passive, where subject and object simply are exchanged: it is a combination of 2–to–1 advancement and 1–to–2 demotion. Call this **exchange**. Exchange in contrast to passive does not create any chômeur. However, to our knowledge such an operation is nowhere attested. There is to our knowledge only one operation where more than one role is involved, namely causatives. Since these are instances of predicate formation, we shall dismiss that case from the present discussion, hoping to resolve its case within a theory of ascensions. Therefore we shall propose the following law:

Single Change Law. Relations may be changed only one at a time.

In the literature as far as we know it there is only one reported case of violation for this law, namely Inversion in Choctaw [24]. However, the relevant change can be analyzed as two changes in succession, namely first 2–to–3 Retreat followed by antipassive. Therefore, the law appears to be universally valid.

We will write a relation change simply as follows: $\alpha \mapsto \beta$, where α and β are relations. [2] This notation is used to refer to a particular indication relation change. However, on many occasions we simply want to state that a relation change has occurred. Then we shall write $[\alpha \mapsto \beta]$ to state that α changes in the next step to β. It may or may not be the case that $\alpha = \beta$.

Using the SINGLE CHANGE LAW we can see that there can be only two operations like the passive: either advancement to 1, leaving the previous subject en chômage, or demotion of subject to 2, pushing the previous object en chômage. The latter kind of operation has been shown to exist. In an analysis of Georgian, Alice Harris ([45]) proposes the following successive changes:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \mapsto \begin{bmatrix} 3 \\ 2 \\ - \end{bmatrix} \mapsto \begin{bmatrix} 3 \\ 1 \\ - \end{bmatrix} \tag{6.10}$$

This sequence of relational changes, called *Inversion*, is like a rochade in chess. The subject retreats to 3 putting the indirect object en chômage. After that the direct object advances to 1.

There is natural tendency (not a law) to favour advancements over demotions. However, RG proposes a law that forbids at least some instances of demotions. A relation is a **term relation** if it is either 1, 2 or 3, otherwise it is a **non–term relation** or an **oblique relation**.

THE OBLIQUE LAW. If β is oblique then $[\alpha \mapsto \beta]$ implies $\alpha = \beta$.

Here are some more laws:

FINAL 1 LAW. At the final stratum, each predicate has a 1.

1 ADVANCEMENT EXCLUSIVENESS LAW. In the course of a derivation, only once per predicate can there be an advancement to 1.

[2]In our case, $\alpha, \beta \in \{1, 2, 3\}$, but this is only a simplification for the sake of exposition. Moreover, the notation is not fully explicit. It does not mention the elements that are involved, but only the relations. As we are exclusively concerned with advancements and demotions, where the predicate remains the same throughout, this is unproblematic.

The apparatus of RG contains also the notion of a dummy (`it` in English, `er` in Dutch, for example), which can fill a grammatical relation. They are needed sometimes to satisfy the FINAL 1 LAW. However, the following must hold. A relation is called **nuclear** if it is 1 or 2.

> NUCLEAR DUMMY LAW. A dummy can only bear a nuclear relation.

Finally we have the

> MOTIVATED CHÔMAGE LAW. If $[\alpha \mapsto -]$ at stratum i only if there is a relational change $[\beta \mapsto \alpha]$ at stratum i.

So, no constituent can put itself en chômage; it must be pushed into chômage by another constituent moving into the relation that the constituent has.

Now once we know how relations are changed, we also need to know how they are assigned. Here, RG assumes that at the initial stratum they are assigned using the θ–grid of the verb. Basically, verbs with identical θ–grid shall end up having identical relation assigned, or more concretely, if a θ–role is assigned GR α with respect to one predicate it shall get role α also with respect to the other. This principle is called the UNIVERSAL ALIGNMENT HYPOTHESIS (UAH). It is stated as follows (originally proposed in [78]; see also the discussion by [82]):

> UNIVERSAL ALIGNMENT HYPOTHESIS. There exist principles of universal grammar which predict the initial relation borne by each nominal in a given clause from the meaning of the clause.

The details are not so well–worked out in the literature, but we shall pick out a particular case. If a verb has an actor, then the actor will always end up bearing the role 1. So an actor is always a deep subject. A theme will end up 2 if an actor is present. The verb `to bite` has an actor and a theme. Hence, the actor is assigned the role 1 and the theme the role 2. In English, 1 is subject and 2 is object. Hence, the final stratum of (6.7) is identical to the initial stratum. The final stratum of (6.8) cannot be the initial stratum since that would violate the UAH. Indeed, the sentence is in the passive voice, and the direct object has been advanced to subject, pushing the former subject into chômeur status.

Figure 6.1: A Lexical Entry

/give/	
$\langle e : \vartriangle\ \varnothing : [\quad]\rangle$	← case–stratum
$\varnothing$	← φ–stratum
$\varnothing$	← GR–stratum
$\langle x : \vartriangle : [\vartheta : agt]\rangle$ $\langle y : \vartriangle : [\vartheta : thm]\rangle$ $\langle z : \vartriangle : [\vartheta : goal]\rangle$	← θ–stratum
$\mathsf{give}'(e); \mathsf{act}'(e) \doteq x;$ $\mathsf{thm}'(e) \doteq y; \mathsf{goal}'(e) \doteq z$	

6.3 Building the Verb: θ–Roles

In this section we will turn to the verb and how it is built up in the lexicon. We shall assume that the basic verb is naked, with as little as possible specified in the various strata. One by one, the argument structure is built up, filling the strata with AISs. We shall assume throughout that verbs come equipped with basically a fragment of their argument structure, which must be filled up by certain controlled processes. It will probably be too much to get into all details of this is achieved, so we need to be content with just glimpses of it.

Verbs are classified by means of some features in order to steer the process of building the argument structure. We shall assume that a verb has features telling us whether it is transitive, benefactive and locational. Furthermore, we need to know whether the subject is experiencer of agent, although that might actually be clear from the verb meaning. We defer the installment of θ–roles to the end of this section and turn to a verb with its θ–grid, A particular lexical entry is shown in Figure 6.1. (For readability, we will insert double line between certain strata.) We can see that the event variable is present only at the case stratum, while the arguments are present only at the θ–stratum. Since the θ–stratum is set like, there is only one argument per θ–role, as we have previously claimed. However, we shall also say that idiosyncratic case marking must be annotated here as well. However, it must be said exactly what is idiosyncratic case marking. According to our views, any case other than nominative (which encodes subjecthood), accusative (which encodes objecthood) and dative (which encodes indirect objecthood) is idiosyncratic, unless it is canonically associated with a θ–role. For example, a beneficiary or an instrument are typically coded in a particular way, which is independent of the verb. As the reader may have noticed already, there are θ–roles in the semantics, and there is an additional θ–stratum into which the θ–roles will be written. We shall call the former kind of θ–roles **semantic θ–roles** and the latter kind **formal**. The semantic θ–roles have the property that they can never be changed, while the formal θ–roles can. Further, there are more semantic θ–roles than there are formal ones. For example, we assume a semantic role of a **mover**, while on the formal side no such θ–role exists.

Now, there are a number of lexical processes that act on the verb in this form. For example, there are a number of verbal prefixes in German that make a verb transitive. For example, the prefix `be-` turns a cofinal locative complement into a direct object.

(6.11) `Johann kletterte auf den Baum.`
`Johann be-kletterte den Baum.`
Johann climbed onto the tree.

In this example, `klettern` (*to climb*) is used with a cofinal PP, which is turned into a direct object by the prefix `be-`. We shall assume therefore that the verb `klettern` has two arguments, an actor and a cofinal location. [3] Thus the argument structure of `be-` is as follows:

/be-/
$\langle e : \Diamond\ominus : [\quad] \mapsto [\quad]\rangle$
$\varnothing$
$\varnothing$
$\langle x : \Diamond : [\vartheta : cof \mapsto thm]\rangle$
$\varnothing$
$\varnothing$

Notice that these prefixes dissociate the abstract θ–role from its original meaning. In the semantics, x is listed as a location towards which some movement takes place, but after applying `be-` it looses its cofinal θ–role and becomes theme instead. However, we shall anyhow not assume that the semantics encodes the θ–roles directly. Rather, what we mean to say when we write $\mathsf{climb}'(e)$ is that there

[3] In order for this proposal to work we need to assume that there are four tpyes of locational θ–roles, one for each mode. This can be motivated by independent reasons.

is an event which consists of someone (or something) performing a certain movement on top of a steep surface. We shall not be concerned here with exactly what it is to climb, but notice that in order to say what it is we must mention at least the actor and the surface. Thus, these two referents will appear explicitly in the semantics, and will also be listed in the θ–grid. The initial assignment of θ–roles can be changed. Moreover, there are a number of θ–roles that can be added more or less freely. One such θ–role is the beneficary. In certain languages the addition of a beneficiary is also marked on the verb. In KinyaRwanda, for example, a suffix `-i` or `-er` is added right after the verbal root if there is a beneficiary. The argument structure of this suffix is therefore as follows:

/-i/
$\langle e : \lozenge\ogreaterthan : [\quad] \mapsto [\quad]\rangle$
$\varnothing$
$\varnothing$
$\langle x : \vartriangle : [\vartheta : ben]\rangle$
$\varnothing$
$\mathsf{ben}'(e) \doteq x$

Notice that this suffix does nothing but add a beneficiary to the θ–grid. KinyaRwanda has another suffix, **-ho**, which is typically at the very end of the verbal suffixes, which adds a locational PP. It is not clear just what its range of meanings is. We have have found it in connection with cofinal and static PPs. Here are two sample sentences, which also show the use of the beneficiary suffix.

	`Íntebe`	`y-iicar-i-w-é-ho`	`umugabo`	`n-uúmwáana.`
(6.12)	chair	it-sit-BEN-PASS-ASP-LOC	man	by-child
	The chair was sat on for the man by the child			

	`Umugabo`	`y-iicar-i-w-é-ho`	`íntebe`	`n-uúmwáana.`
(6.13)	man	he-sit-BEN-PASS-ASP-LOC	chair	by-child
	The man was-sat-on-the-chair for by the child			

The fact that the location can be passivized shows that the locative suffix actually turns the location into a direct object. Hence, for reasons that shall become clear we shall assume that the locative suffix does actually not add the location to the θ–grid but rather advances it to direct object. The verb sit obviously has a locational complement, but it does seem that the locative suffix is appropriate even when a verb does not have a locative complement. Hence, we shall assume that a verb can take a location, and this is marked by a zero suffix, which takes one of the four forms COF, COIN, TRANS or STAT:

$$(6.14)\quad \begin{array}{c} /\textsc{cof}/ \\ \hline \multicolumn{1}{|l|}{\langle e : \Diamond\ominus : [\quad] \mapsto [\quad]\rangle} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\langle x : \vartriangle : [\vartheta : cof]\rangle} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\mathsf{move\text{-}to}'(e, y, x)} \\ \hline \end{array}$$

Notice that the presence of the locative complement is signalled by the feature [LOC : ±]. Further, in the semantics appears an additional variable, y, which needs to be construed. It is typically the actor or theme, but we shall ignore that problem here.

We shall return now to the problem of installment of θ-roles. Since θ–roles have meaning, it should in fact not be so problematic to install a θ–role by a canonical process of hooking something by its description.

$$(6.15)\quad \begin{array}{c} /\textsc{agt}/ \\ \hline \multicolumn{1}{|l|}{\langle e : \Diamond\ominus : [\quad]\rangle} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\langle x : \vartriangle : [\vartheta : agt]\rangle} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\mathsf{act}'(e) \doteq x} \\ \hline \end{array}$$

This approach works as soon as we assume the verb to supply a canonical argument that is actor; for example, if the verb carries in its semantics a statement of the form $\mathsf{act}'(e) \doteq y$, then $y \doteq x$ can be inferred immediately, and so x is linked to y by a canonical logical inference. Otherwise, if nothing is specified then the general meaning of the verb must make it clear which is the actor. However, notice that there are a number of cases where this approach does not work properly. A particularly well–documented case is the so–called transitive alternation.

(6.16) `Alfred loaded the hay onto the truck.`

(6.17) `Alfred loaded the truck with hay.`

Figure 6.2:

/load$^{\checkmark}$/

$\langle e : \vartriangle \ \varnothing : \star \rangle$
$\varnothing$
$\varnothing$
$\varnothing$
e
$\mathsf{give}'(e, x, y, z)$; $\mathsf{act}'(e) \doteq x$;

The verb `to load` takes three arguments, and agent, a location and an object. The latter is typically called the theme. We may however also call the location the theme and then the former theme is the instrument. So, the verb appears in either of the following configurations:

$$
(6.18) \qquad \begin{array}{lll} x : \text{actor} & y : \text{theme} & z : \text{location} \\ x : \text{actor} & z : \text{theme} & y : \text{instrument} \end{array}
$$

Two approaches are conceivable. The first considers one construction as basic and the other as derived, and the second considers both as concurrent realizations of the same verb. These solutions are readically different in the way they conceptualize the installment of θ–roles. In the first view we accept a basic assignment of θ–roles and derive the alternative patterns through a series of empty morphemes, and in the second view we see the assignment of θ–roles as basically flexible, determined in large parts by the meaning of the verb.

The two views may be reconciled by accepting that for some but not all θ–roles a choice is available. We shall say, for example, that in the semantics it is explicitly stated that x is the actor but that nothing is specified for y or z. The verb is explicitly marked for transitivity. This determines that either of y or z is the direct object. The lexical entry is shown in Figure 6.2. Notice that we have now written explicitly $\mathsf{give}'(e, x, y, z)$. This means that e is an event of loading, where loading is an activity taking an actor, a substance and a location. Furthermore, it is explicitly said that x is the actor, while we know nothing about y and z. We imagine that `load` has optionally the feature $[\textsc{cof} : +]$. This feature can be reset in English by an empty morpheme, which we shall call TA.

(6.19)

/TA/

$\langle e : \Diamond\ominus : \begin{bmatrix} \text{COF} & : & \star \mapsto + \\ \text{TRS} & : & \star \mapsto + \end{bmatrix} \rangle$
$\varnothing$
$\varnothing$
$\langle x : \vartriangle : [\vartheta : thm] \rangle$
$\varnothing$
move-to′(e, y, x)

So, when the cofinality feature is positive (indicating a goal of the movement), then it can be set to $\star$ in tandem with the transitivity by TA. Indeed, after TA has applied we have a transitive verb. It remains to add the θ–role of the third argument. There is nothing that guides us here. We shall assume here that the choice of instrument is semantically determined. Similarly, if the other option is taken. Then the verb is not a locational verb but transitive. We shall assume that the default choice is to take the substance as the theme. The precise meachnics of this proposal depends in large parts on a fine grained analysis of θ–roles, or to be exact, in pre–θ–roles. In the present example, we have an actor, a location and a substance. We shall assume for example that substance is taken to be theme if nothing else is specified. [4] The element TA is actually quite similar to German `be-`. In fact, the two examples must be translated into German as follows.

(6.20) `Alfred loaded the hay onto the truck.`
`Alfred lud das Heu auf den Laster.`

(6.21) `Alfred loaded the truck with hay.`
`Alfred be-lud den Laster mit Heu.`

It seems however that `be-` is slightly more productive than TA. Moreover, `be-` can turn an intransitive verb into a transitive one. If we analyse `be-` as acting even before θ–roles are installed, we shall say that it can set the transitivity value of the verb from $-$ or $+$ to $\star$ (thus accounting for the fact that it applies as well to intransitive verbs) in addition to changing COF from $+$ to $\star$.

This ends the discussion of θ–role installment. What may have become clear is that there is a subtle interaction between the meaning of the verb, the meaning of

[4]However, it needs to be seen how such preferences or defaults can be built in or whether they can be stated in such a way that they are not defaults but rules.

the θ–roles and certain classificatory features indicating the presence of a theme, a location or a beneficiary. These features are mainly the device to control the installment of θ–roles.

6.4 Building the Verb: Grammatical Relations

In the previous section we have discussed how the θ–roles are installed into the argument structure of the verb. Here we will focus on the grammatical relations, and finally say something on agreement. Once the θ–roles have been assigned, there is a process of filling the GR–stratum with AISs. There are at most three of them: 1, 2, and 3. This will account for the structure of the initial stratum. The assumption behind our proposal is the following specific variant on the UAH:

> At the initial stratum, the grammatical relations 1, 2 and 3 correspond each to a specific set of θ–roles, which are mutually disjoint.

In other words, for each θ–role there exists at most one GR to which it initially belongs. There are clear and less clear cases. The theme is invariably 2, while actors are always 1s. Experiencers will also count as 1s. The fact that the experiencer often surfaces as a 3 will have to be explained. Goals are 3s. Many θ–roles do not have a GR assigned to them, which means that they will end up bearing no GR at the initial stratum.

By means of some (typically empty) elements, the θ–roles get excanged for a GR. These elements are called 1INST, 2INST and 3INST. Below we show 1INST and 2INST.

(6.22)

/2INST/	/1INST/
$\langle e : \lozenge\ \circlearrowleft : [\quad] \rangle$	$\langle e : \lozenge\ \circlearrowleft : [\quad] \rangle$
$\varnothing$	$\varnothing$
$\langle x : \vartriangle : [\textsc{gr} : 2] \rangle$	$\langle x : \vartriangle : [\textsc{gr} : 1] \rangle$
$\langle x : \triangledown : [\vartheta : \mathit{thm}] \rangle$	$\langle x : \triangledown : [\vartheta : \mathit{act} \sqcup \mathit{exp}] \rangle$
$\varnothing$	$\varnothing$
$\varnothing$	$\varnothing$

As one can see, the installment elements take away the θ–role in exchange for a grammatical relation. This is a way to see to it that the θ–roles do get installed as

Figure 6.3: GR–Installment

/`give`/
$\langle e : \vartriangle\ \varnothing : [\quad] \rangle$
$\varnothing$
$\varnothing$
$\langle x : \vartriangle : [\vartheta : agt] \rangle$ $\langle y : \vartriangle : [\vartheta : thm] \rangle$ $\langle z : \vartriangle : [\vartheta : goal] \rangle$
e
$\mathsf{give}'(e, x, y, z); \mathsf{act}'(e) \doteq x;$ $\mathsf{thm}'(e) \doteq y; \mathsf{goal}'(e) \doteq z.$

$\bullet$

/3INST/
$\langle e : \lozenge\ \circlearrowleft : [\quad] \rangle$
$\varnothing$
$\langle x : \vartriangle : [\textsc{gr} : 3] \rangle$
$\langle x : \triangledown : [\vartheta : goal] \rangle$
$\varnothing$
$\varnothing$

$=$

/`give` + 3INST/
$\langle e : \vartriangle\ \varnothing : [\quad] \rangle$
$\varnothing$
$\langle z : \vartriangle : [\textsc{gr} : 3] \rangle$
$\langle x : \vartriangle : [\vartheta : agt] \rangle$ $\langle y : \vartriangle : [\vartheta : thm] \rangle$
e
$\mathsf{give}'(e, x, y, z); \mathsf{act}'(e) \doteq x;$ $\mathsf{thm}'(e) \doteq y; \mathsf{goal}'(e) \doteq z.$

grammatical relations. That is to say, the installment of GRs is obligatory. We will see that there are other ways to control the installment. By the fact that θ–roles have only one GR assigned to them, a verb never has more than one particular GR, or to be exact, the GRs do not cancel each other at the installment phase. However, as we have already said, there are numerous θ–roles that do not end up with a GR.

After the installment of GRs, valency changing operations can (and sometimes must) apply. We have spoken already in Section 3.4 about such operations. We have noted there that there are far more operations than just passive. There are languages which can promote various actants to subject. In this section we shall closely analyse the mechanics of these operations. Let us take the ordinary passive first. According to Relational Grammar, the English passive promotes the object to subject. As a side effect, however, the former subject is removed from its position and becomes a chômeur. This is neatly analysed in our system as follows. Passive has the following argument structure.

(6.23)

/PASS/
$\langle e : \Diamond \curvearrowleft : [\text{VOICE} : \star \mapsto \textit{pass}]\rangle$
$\varnothing$
$\langle x : \Diamond : [\text{GR} : 2 \mapsto 1]\rangle$
$\varnothing$
$\varnothing$
$\varnothing$

If passive applies to a verb that has a subject, then since the GR–stratum is a set, one of the relations has to go. We shall assume here (as always) that the functor wins. In this case, the passive morpheme is the functor, and its subject will be the subject of the complex V+PASS. It follows that the former subject shall be without a GR. An example is shown in Figure 6.4. A subject chômeur is no longer a subject, but it can nevertheless be an actant of the sentence.

(6.24) `Ernie was hit by Bert.`

(6.25) `The book was given to Jane by Bill.`

In Relational Grammar this is accounted for assuming a typology of chômeurs. For each GR there is a corresponding chômeur. So, a subject which is pushed en chômage is a subject chômeur and hence distinct from an object that is pushed en chômage. If this view is correct, then the subject does not loose its grammatical relation but exchanges it for another. Now, as we shall argue, this view is not unproblematic. Theory internally it is of course contradictory to define a chômeur as someone who has just lost his GR, and at the next step define a new relation of a chômeur. Surely it would be preferrable if it were really the case that the chômeurs really bear no grammatical relation. This is actually what we will assume here. Notice first of all that even though the subject chômeur has lost its GR, it is nevertheless listed in the semantics as being an actor. Therefore, we shall assume that it is possible to add an adjunct by–phrase to specify the actor of an event. If this view is correct, then the availability of a by–phrase is not dependent on the prior existence of a subject but rather on the prior existence of an initial subject, because the latter is defined in terms of semantical functions. To see this, notice that there are verbs whose subject is not an actor. One such example is

Figure 6.4: Passive in English

/$\texttt{hit}$/

$\langle e : \triangle \ \varnothing : [\textsc{voice} : \star] \rangle$
$\varnothing$
$\langle x : \triangle : [\textsc{gr} : 1] \rangle$ $\langle y : \triangle : [\textsc{gr} : 2] \rangle$
$\varnothing$
e
$\mathsf{hit}'(e, x, y); \mathsf{act}'(e) \doteq x; \mathsf{thm}'(e) \doteq y.$

$\bullet$

/$\textsc{pass}$/

$\langle e : \Diamond \curvearrowleft : [\textsc{voice} : \star \mapsto \textit{pass}] \rangle$
$\varnothing$
$\langle x : \Diamond : [\textsc{gr} : 2 \mapsto 1] \rangle$
$\varnothing$
$\varnothing$
$\varnothing$

$=$

/$\texttt{hit} + \textsc{pass}$/

$\langle e : \triangle \ \varnothing : [\textsc{voice} : \textit{pass}] \rangle$
$\varnothing$
$\langle x : \triangle : [\textsc{gr} : 1] \rangle$
$\varnothing$
e
$\mathsf{hit}'(e, x, y); \mathsf{act}'(e) \doteq x; \mathsf{thm}'(e) \doteq y.$

German bekommen (*to get*). There are also verbs, reuen (*to repent*), which have an accusative experiencer. None of the verbs tolerate a by–phrase in the passive. Moreover, they do not even like to be passivized at all:

(6.26) Ich bekomme das Buch.
I get the book.

(6.27) ?Das Buch wird bekommen.
The book is got.

(6.28) ?Das Buch wird durch mich bekommen.
The book is got by me.

(6.29)

Now, if the subject matter is actually the final subject, why can it not be passivized? We claim that this is so since passive in German can only apply to initial 2s. This might well be different in other languages. Furthermore, the former subject can always be added in the form of a PP headed by von. We may therefore assume that von has — among other — the following argument structure.

(6.30)

/von/
⟨*e* : ◊⧁ : [VOICE : *pass*]⟩ ⟨*x* : ▽⧁ : [DEF : ⊤]⟩
∅
∅
∅
∅
act′(e) $\doteq x$

In this way, if von applies to an NP, it returns a verbal adjunct that specifies the actor of the verb, but only if the verb is put into the passive. This analysis can be extended to objects and indirect objects as well. Notice that the present analysis has a drawback: it does not allow to control for the number of von–PPs and likewise for English by–phrases. If it is an adjunct, then it can be freely added, but it can also be added twice. So, the following sentence is predicted to be grammatical, contrary to fact.

(6.31) ?Ernie was hit by Bert by Kermit.

There is a solution that offers itself immediately: we shall take it that the installment of GRs does not lead to the consumption of θ–roles. Although that is not unproblematic in itself as we have remarked above, it does lead to a correct blocking of (6.34). However, now we must account for the fact that the by–phrase can be freely omitted.

We have already seen verbs which even in the active voice do not show an exact correspondence between θ–roles and GRs. It is assumed in RG that such verbs have already undergone relational change, so that what we see is not their initial stratum but a higher stratum. An example are experiencer verbs in German (and many other languages). Often, their subject is in the dative for example with scheinen (*to seem*), though the accusative is also possible as we have seen above. With these verbs, the subject matter (that which is experienced) is the subject. If the experiencer is the final subject, we must assume that a number of relational changes have applied. Indeed, we shall assume that the initial subject first retreats to 3 and the subject matter advances to 1. If the initial subject is the final object, matters are more complex. One compatible analysis is when the subject retreats to 3, after which the subject matter is advanced to 1. Finally, the 3 is readvanced to 2. By the 1–Advancement Exclusiveness Law there can be no passive, even though the verb is actually morphologically in the active.

(6.32) Diese Sache reut mich.
I repent this.

(6.33) *Ich werde gereut.
I am being repented.

Now, we see that sometimes changes are prohibited, sometimes free and sometimes mandatory. To account for this, verbs need to be given certain features. For example, we need to mark certain verbs with a feature that tells us that the verb must undergo inversion. To make this work, we require that relational change is sensitive to these features, and that certain features must be obligatorily cancelled before any other operation can apply. We shall not go into the details here.

We shall now turn to a set of operations that is distinct from the previous ones. These are operations that promote a certain θ–role to a grammatical relation which is not canonically associated with it. A case in point is the passive and the particle -ho in KinyaRwanda. As is argued by Dryer in [29], passive promotes not only 2 to 1, but also 3 and beneficiaries. So, since beneficiaries are not canonical 3s, we shall assume that passive has a different function in the case of beneficiaries, namely, it installs them into the GR–stratum. Other than that, it works just as ordinary passive. The argument structure is shown in Figure 6.5. Likewise, -ho

Figure 6.5: Beneficiary Advancement

/BEN–1/

$\langle e : \Diamond \curvearrowleft : [\textsc{voice} : \star \mapsto \mathit{pass}]\rangle$
$\varnothing$
$\langle x : \vartriangle : [\textsc{gr} : 1]\rangle$
$\langle x : \triangledown : [\vartheta : \mathit{ben}]\rangle$
$\varnothing$
$\varnothing$

is advancement of a location to 2. Namely, as can be gathered from the data in Section 6.3, the locative argument can be passivized if `-ho` is suffixed to the verb. Moreover, it appears that if a verb carries the suffix `-ho` then it is the initial location that must be the subject, nothing else can be. It is not clear to us why this is so. Presumably, verbs in which a location is promoted to direct object have a special status.

To conclude this section we take a look at agreement. After the grammatical relations have been assigned and changed, agreement is added onto the verb. The basic idea is that the agreement markers appear in a fixed order (either being prefixes or suffixes), starting inside out with 3, then 2 and last 1. The basic argument structure has been laid out in Chapter 3. Here we shall adress a question of detail that is quite crucial for the next section. For a smooth layering of the agreement suffixes one might simply assume that the agreement markers remove the grammatical relation. AgrIO then takes the 3 and adds the case and φ–features of the indirect object. AgrO takes the 2 and adds the case and φ–features of the direct object. Moreover, AgrO cannot apply if a 3 is still present. Finally, AgrS applies, removing the 1 and adding case and the φ–features of the subject. AgrS can apply only if no 2 and no 3 is present. If matters are done in this way, however, two facts are apparently missed:

1. There are number of languages in which agreement extends to arguments without a grammatical relation (for example beneficiaries).

2. Numerous constituents of the sentence must be construed with an argument of the verb, and the grammatical relation is quite often the decisive criterion

for linking. If the grammatical relation is absent, then linking cannot be properly defined.

Therefore, we may alternatively assume that agreement does *not* remove the grammatical relation. This may at first seem to remove the basis for a proper control of the agreement features. But this is not so. Notice namely that since agreement adds an AIS to the φ–stratum and the case stratum, it is possible to detect whether or not agreement has applied. So, we may say that AgrDO can only apply if there is either no 3 or else a dative argument is present at the case and the φ–stratum. It is admittedly somewhat awkward to spell out the correct argument structure, so one may alternatively resort to adding some features controlling the assignment of the agremeent suffixes. Nevertheless, in the next section we shall look at data that will show that we must assume agreement to remove the GR.

6.5 Linking I

In this section we shall look at constructions that involve what is sometimes called *linking* and sometimes also *construal*. When two constituents meet they are more often than not exchanging more than one referent. Typical examples are PPs.

(6.34) `John saw Mary at the corner.`

(6.35) `John moved the furniture to the other wall.`

(6.36) `John saw Mary with a handbag.`

In each of these cases we must decide to whom the PP applies, so to speak. In (6.34), we may read this sentence as saying that John saw Mary, and *she* was at the corner at that moment. Or we can read it as saying that John saw Mary and *he* was at the corner at that moment. We say that the PP is in the first case be construed with the object and in the second case with the subject. Generally, PPs are different with respect to the possibilities of construal. For example, the PP `to the other wall` cannot be construed with the subject, only with the object. In (6.36) it too seems that construal with the object is obligatory, but choice of a different verb (`hit`) reverses this intuition.

We shall say that the argument (or referent, for that matter) with which the PP needs to be construed is the **target** of the PP. Now, in each of these cases it must be carefully analysed what parameters determine the choice of target. At least two factors come into play for the choice of the target:

1. the semantic θ–role of the target
2. the grammatical relation of the target

There are additional factors such as the particular meaning of the verb or the arguments, but we shall exclude them from the discussion here. Another obvious candidate are the formal θ–roles. Yet, formal θ–roles are not at our disposal, since they are systematically discharged. However, notice that θ–roles do survive as semantic θ–roles and so it does seem that the discharge of formal θ–roles actually has no bearing on the ability to trigger construal. Moreover there does not seem to be a differenec between construal based on formal θ–roles and construal based on semantic θ–roles. We shall demonstrate however that there is a difference between these options. For if the formal θ–roles trigger construal then we should expect some sensitivity to lexical processes which change the assignment of formal roles. A case in point is the role of target of a directional PP. As is argued in [59] the target of a directional PP is always the mover, and the latter is a semantic θ–role, not a formal one. Moreover, the mover need even not be realized as an overt argument. Let us take the verb `bohren` (*to drill*). In the following sentences, the scene is always the same: the drill is going from above through the steel. There are two directional PPs, `durch den stahl` (*through the steel*) and `von oben` (*from above*).

(6.37) `Der Bohrer bohrte sich von oben durch den Stahl.`
The drill was drilling from above through the steel.

(6.38) `Alfred bohrte (mit dem Bohrer) von oben durch den Stahl.`
Alfred was drilling (with the drill) from above through the steel.

(6.39) `Alfred durchbohrte (mit dem Bohrer) von oben den Stahl.`
Alfred was through–drilling (with the drill) from above through the steel.

Now, `the drill` is the agent subject of (6.37), and the instrument in (6.38) and (6.39). Moreover, it is only optionally present in the latter two. There is no choice for the target of construal in any of these constructions. If we choose the (formal) agent as target, then we get that Alfred must also be a target of construal in (6.38) and (6.39). This is not the case. If we choose the formal instrument as target, then in (6.40) below construal of the PP `ins Tor` with `Alfred` is possible, contrary to fact. For the sentence cannot mean that Alfred ends up in the goal as a result of shooting the ball.

(6.40) `Alfred schoß den Ball mit dem linken Fuß ins Tor.`
Alfred shot the ball with his left foot into the goal.

These observations are rather stable across all languages. A directional PP must be construed with the mover, no other characterisation will do. The *mover* is however a semantic θ–role not a formal one. Moreover, construal is stable under any lexical process and this is evidence for the fact that a characterization in terms of formal θ–roles is inadequate. If this is so, the lexical entry for `durch` (in its locative meaning) is as follows:

(6.41)

/durch/
$\langle e : \Diamond\ominus : [\quad]\rangle$ $\langle x : \triangledown\ominus : [\textsc{case} : acc]\rangle$
$\langle x : \Diamond : [\textsc{def} : \surd]\rangle$
$\varnothing$
$\varnothing$
$\varnothing$
$\text{move-to}'(e, y, x); \text{mov}'(e) \doteq y.$

Notice that the proper construal does not only need the referents x and y but also the event. One can similarly argue that other PPs depend in their interpretation not on the formal θ–roles or grammatical relations but rather on the semantical θ–roles. Instruments are those employed by agents, therefore the one using the instrument must be the agent. A particularly interesting case is `with`, which forms either an instrument or indicates possession. If used in the latter sense it can be construed with the object, but not if used in the former sense.

(6.42) `John hit the dog with a stick.`
`The dog was hit (by John) with a stick.`

(6.43) `John saw Mary with a telescope.`
`Mary was seen (by John) with a telescope.`

In (6.42) `with a stick` is an instrument if construed with `John`, and a possessive if construed with `the dog`. This does not depend on whether the sentence is active or passive (hence showing that grammatical relations are not involved either). Similarly with (6.43). If we replace `with a stick` by `with a smile` not only does the instrumental reading vanish (that the smile is instrumental in hitting or seeing is rather implausible) but also the construal with the subject becomes more plausible.

There are numerous other constructions that involve linking of one or the other sort. Such are reflexives and infinitives. We shall say a little bit about infinitives. This will be a good opportunity to see how we have progressed from the results of Section 2.8. In that section we described control as the result of linking to a particular argument of the lower verb. This argument was identified by means of θ–roles. It is known that this is inadequate. Rather, linking can only be with the subject of the infinitive regardless of its θ–role. Hence, the assumptions of Section 2.8 concerning the mechanism of linking are questionable. Here we shall propose an alternative based on GRs. Notice that any control verb can either take an infinitival or a finite clause as a complement.

(6.44) `John persuaded Mary to go to London.`

(6.45) `John persuaded Mary that they should go to London.`

(6.46) `John promised Mary to go to London.`

(6.47) `John promised Mary that they would go to London.`

The difference between the finite and the infinite complements is that the latter require identity between the lower subject and the controller. The controller is the agent (as with `promise`) or the patient (as with `persuade`). The controller is not identified by its GR. For example, we have subject control in (6.48).

(6.48) `Mary was persuaded (by John) to go to London.`

Promise does not like to be used with the infinitive in the passive for reasons that are unclear to us. Now, how do we account for these facts? Clearly, we can declare in the semantics of the control verb in which way it controls the lower verb:

(6.49)

$$\begin{array}{l} \texttt{/persuade/} \\ \hline \multicolumn{1}{|l|}{\langle e : \vartriangle \ \varnothing : [\quad] \rangle} \\ \hline \multicolumn{1}{|l|}{\varnothing} \\ \hline \multicolumn{1}{|l|}{\langle y : \Diamond : [\textsc{gr} : 1 \mapsto 2] \rangle} \\ \hline \multicolumn{1}{|l|}{} \\ \hline \multicolumn{1}{|l|}{e} \\ \hline \multicolumn{1}{|l|}{\mathsf{persuade}'(e, x, y, e');} \\ \multicolumn{1}{|l|}{\mathsf{agt}'(e) \doteq x; \mathsf{pat}'(e) \doteq y.} \\ \hline \end{array}$$

Notice that as for y, no θ–role installment is necessary or even allowed. We shall assume that it comes out of the lexicon with the GR assigned to y. Now, by the rules of discharge, y cannot be fully discharged by object agreement. The fully inflected verb nevertheless imports y under the subject GR. However, when the verb is combined with its infinitival complement, it can link y with the missing subject of the infinitival complement. The Figure 6.6 shows the control of a subject of an infinitive.

At this point we can see why it is desirable to let agreement take away the GR. For if it were to leave the GR, the control verb could still link with the subject of the lower clause even if the latter is finite. Hence we would have to state explicitly that if the complement is finite, no linking can take place and if it is infinitival, linking is mandatory. But if the finite clause has no GR 1, then linking is impossible anyway. So, by leeting agreement take away the GRs we do not need to account for the fact that the complement is infinitival if linking takes place: there is no other way to do it. We should also say here explicitly that we assume infinitives of German and English to assign case to their objects, so that an infinitive has object agreement, but no subject agreement.

There is a last detail to be mentioned, namely the status of the complement infinitive. We shall assume first of all that all complement clauses are marked for accusative. However, if there is an additional object NP, then the most immediate complement is the NP, and not the infinitive. Moreover, we can passivize for the NP and not the complement in this case. Therefore, the complement clause has the status of an object chômeur.

To be distinguished from control predicates are the predicates that trigger clause union. In German these are `lassen`, `helfen`, `wollen` and more. Their

Figure 6.6: Control

/John persuades Mary/

$\langle e : \triangle\ \varnothing : [\textsc{vform} : \textit{fin}]\rangle$ $\langle e' : \nabla$⧁$: [\textsc{vform} : \textit{inf}]\rangle$
$\langle x : \lozenge : [3.\textsc{sg.masc}]\rangle$ $\langle y : \lozenge : [3.\textsc{sg.fem}]\rangle$
$\langle y : \nabla : [\textsc{gr} : 1]\rangle$
$\varnothing$
$e, \quad j, \quad m, \quad e'$
persuade$'(e, x, y, e')$; agt$'(e) \doteq x$;   pat$'(e) \doteq y$; sub-matt$'(e) \doteq e'$.

$\bullet$

/to sing/

$\langle e : \triangle\ \varnothing : [\textsc{vform} : \textit{inf}]\rangle$
$\varnothing$
$\langle x : \triangle : [\textsc{gr} : 1]\rangle$
$\varnothing$
e
sing$'(e, x)$; agt$'(e) \doteq x$.

$=$

/John persuades Mary to sing/

$\langle e : \triangle\ \varnothing : [\textsc{vform} : \textit{inf}]\rangle$
$\langle x : \lozenge : [3.\textsc{sg.masc}]\rangle$ $\langle y : \lozenge : [3.\textsc{sg.fem}]\rangle$
$\varnothing$
$\varnothing$
$e, \quad j, \quad m, \quad e'$
persuade$'(e, x, y, e')$; agt$'(e) \doteq x$; pat$'(e) \doteq y$;   sub-matt$'(e) \doteq e'$; sing$'(e', y)$; agt$'(e') \doteq y$.

syntax has been discussed extensively in Section 2.8 and also in Section 3.7. There we have claimed that in distinction to English, these verbs have a different argument structure in Dutch and German: the infinitive is the first argument rather than the last. The consequence of this fact is that these verbs form clusters, which can have arbitrary size. The different syntax needs accounting for. The problem is that the argument structure must be built up one by one. Moreover, for any verb that needs an infinitival complement to be discharged first (as is the case in German and Dutch) we have a problem in assigning the right analysis with respect to agreement. For we must assume that case assignment is added to the case–stratum always immediately after the event variable. For the case–stratum must eventually contain the following sequence:

$$e\text{: predicate, } x\text{: subject, } y\text{: object, } e'\text{: complement predicate}$$

But before the agreement markers are added we have

$$e\text{: predicate, } e'\text{: complement predicate}$$

Hence, if first y is added then x and if the place of insertion is the place right after e, then we indeed get the right argument structure. The complement is the first referent that needs to be discharged. In this way we get the fact that verbs form a cluster. Now, if we insert simply at the end, that is after the e' variable, then we get the following argument structure:

$$e\text{: predicate, } e'\text{: complement predicate, } x\text{: subject, } y\text{: object}$$

Hence, the distinction between these two constructions lies only in the insertion point of the nominal referents. English presents an intermediate case. Here the target sequence is

$$e\text{: predicate, } x\text{: subject, } e'\text{: complement predicate, } y\text{: object}$$

To implement this, we shall assume that fusional adjuncts can determine how the resulting argument structure will look like. Now, an adjunct is fusional if it itself has arguments which are unsaturated. The question is therefore where these arguments are being put. They can be put either at the beginning of the sequence (which take to be the second position) or right at the end. We mark the first option by a dot before the horizontal diacritic, and the second by a dot following the diacritic. So we have either $\langle e, .\blacklozenge\mathfrak{b}, \mathfrak{z}\rangle$ or $\langle e, \blacklozenge.\mathfrak{b}, \mathfrak{z}\rangle$. As a case in point we take the agreement suffixes. They are or the required kind: they are adjuncts of their event variable, and they introduce an argument into the case–stratum. We assume that German and Dutch agreement suffixes are of the following kind:

(6.50)
$$\begin{array}{lcl} \text{AGRS} & : & \langle e, .\blacklozenge\mathfrak{b}, \mathfrak{z}\rangle \\ \text{AGRDO} & : & \langle e, .\blacklozenge\mathfrak{b}, \mathfrak{z}\rangle \end{array}$$

English differs with respect to the object:

(6.51)
$$\begin{array}{lcl} \text{AGRS} & : & \langle e, .\blacklozenge\mathfrak{b}, \mathfrak{z}\rangle \\ \text{AGRDO} & : & \langle e, \blacklozenge.\mathfrak{b}, \mathfrak{z}\rangle \end{array}$$

In this way we can generate different serializations for the arguments of the verb.

Predicates which trigger clause union also have in some languages the property to take up the argument grid of the lower verb. This has been described for Spanish by [2], and for Hungarian in Section 3.5. The verb `akarni` in Hungarian takes object agreement if construed with a transitive verb. An explanation for these facts consists in assuming that there are two kinds of infinitives: the first infinitive assigns case only to its non–subject arguments but only a GR to its subject, and the second, which assigns no case and only GRs to its arguments. Call the second type **bare infinitive**. In English and German, bare infinitives can be distinguished from infinitives by the fact that the latter take `to` in English and `zu` in German. Lets assume that Hungarian has bare infinitives. Then a verb can select a bare infinitive. Since this infinitive does not assign at all to its argument, this must be done by the matrix verb. There is in fact nothing required for this possibility to exist. Simply note that we can attach direct object agreement to any verb, transitive or not. To see this one just has to take a look at the representation (Figure 6.7). It this structure is merged with that of an intransitive verb, we get a verb that is looking for an argument with a 2. This in turn exists only if there is a bare infinitive around that supplies a 2. So if the verb is a raising verb, and it selects a bare infinitive, then we get transitive agreement. Moreover, this agreement is obligatory.

This solution works fine for Hungarian, but what about other languages? [2] discuss clitic climbing in Spanish. They claim that what is at issue is not that a clitic is climbing upstairs but that two predicates form a complex predicate (and

Figure 6.7: Object Agreement Suffix (Hungarian)

/AGRO/

$\langle e : .\blacklozenge⧀ : [\text{FORCE} : \checkmark]\rangle$ $\langle x : \triangledown⦶ : [\text{CASE} : nom]\rangle$
$\langle x : \Diamond : \begin{bmatrix} \text{PERS} : \pi & : \\ \text{CLASS} : \kappa & : \\ \text{NUM} : \nu & : \end{bmatrix}\rangle$
$\langle x : \triangledown : [\text{GR} : 2]\rangle$
$\varnothing$
$\varnothing$
$\varnothing$

hence that there is one clause and not two). Their evidence is the fact that clitics may or may not climb, but if one is climbing, all others must do so as well. This means that the choice of whether or not a clitic is climbing is not one that the clitic alone can make. Rather, the whole construction does so. Therefore, if we assume that the verbs may or may not fuse, and therefore may or may not form a single predicate, then this is easily explained. These facts are fall into the framework here; and they give additional support for the thesis by Perlmutter and Aissen. For the facts of Hungarian and Spanish are quite similar, only that Spanish has clitics where Hungarian has agreement suffixes. So all we need to do is to assume that these clitics function like agreement markers. Additional support comes from raising to subject. Notice that our theory predicts that English and German can have matrix verbs agree with lower subjects. (It predicts that for objects too (in case there exist bare infinitives selected by verbs as in German) but there is no way to test this prediction.) For notice that even if a verb like `to seem` selects a normal infinitive (hence not a bare infinitive), this infinitive does not assign case to its subject. So the matrix verb can and must show agreement with the subject of the lower infinitive. This subject must appear to the left hand side of the matrix verb. This generates exactly the right facts.

(6.52) `You seem to be clever.`

(6.53) *`Seem you to be clever.`

(6.54) *`Seem to be clever.`

(6.55) `It seems to rain.`

(6.52) is grammatical, since AGRS has attached to the verb `seem`, forming a verb looking for a 1. `to be clever` is an infinitive, which supplies it. (6.53) is illegitimate since if AGRS attaches to the matrix verb, the subject must be to the left. (6.54) is out because there is a 2 that is not consumed. (6.55) is grammatical, since there is also a 'dummy' AGRS, which applies when no 1 is present.

Notes on this section. If our theory of complex predicates is correct then it actually follows that complex predicates can only be built so long as there we do not get the same GR twice. Hence, this construction is generally limited to a small class of serial verbs. This seems to be the case for Hungarian. Whether or not this holds for German or Dutch must be questioned. Hence, we must assume that what the Dutch and German verbs fuse with is an ordinary infinitive and not a bare infinitive.

6.6 Linking II

In this section we shall discuss the linkage of clauses. We shall look at specific devices that allow to link the arguments of different clauses with each other. Such devices are called **reference tracking systems**. These systems are quite diverse and therefore this treatment cannot be exhaustive. The next chapter will actually also deal with reference tracking altough by means of pronouns. Here, we shall be concerned with nonpronominal systems. We shall an example. Martuthunira is a language of Western Australia (see [27]). Like many other Australian languages, Martuthunira has a special kind of clause, the purposive clause. This is an adjunct clause specifying for what purpose an activity is performed. In a purposive clause the subject is often omitted. Instead, the verb in Martuthunira carries either of three suffixes:

(6.56)

`-lu/-ru` (PURPSS)	subject of purpose clause is nondisjoint with subject of main clause
CM-`WAA` (PURPS=O)	subject of purpose clause is nondisjoint *with*an accusative object of main clause
-CM-`WALA` (PURPDS)	subject of purpose clause is disjoint with subject of main clause

(Here, CM means 'conjugation marker'. This is a suffix depending on the conjugation class. There is no agreement with either subject or object.) The use of same subject marker is illustrated by the following examples:

(6.57)

`Kayarra`	`kanarri-lha`	`nganaju`	`nhawu-lu`
two	come-PAST	1.SG.ACC	see-PURPSS

Two people came to see me.

(6.58)

`Kartu`	`puni-layi`	`minthal-wa-rru`	`nhuwa-ru`	`jankurna-a`
2SG.NOM	go-FUT	alone-∅-NOW	spear-PURPSS	emu-ACC

You can go alone now to spear an emu.

Here, nondisjoint means that if the subjects are individuals they are different, and if they are groups then the two groups are not disjoint. (So, if one is an individual and the other a group, then the group does not contain the individual.) Dench uses the word 'coreferential', but that is improper usage of terms. Hence, PURPSS is put even when the subjects of the two clauses are different but have at least one member in common. We shall return to this further below. Now, here is an example of the use of PURPS=O:

(6.59)

`Ngayu`	`kartungu`	`parla-marta`	`purra-rninyji`	`pal.ya-a`
1.SG.NOM	2SG.ACC	stone-PROP	hit-FUT	temple-ACC
`pungka-waa-rru`				
fall-PURPS=O-NOW				

I'll hit you with a stone in the temple so you fall down.

Martuthunira marks the direct object, the indirect object as well as the benefactive by accusative. All three can be used as controllers of purposive clauses. However, while it is anyway rare for two accusative marked complements to appear in one sentence, it is impossible for them to head each a separate purposive clause.

Finally, the PURPDS marker signals that the subjects of the two clauses are different. It is stated in [27] that if PURPS=O can be used, it is preferred, so that PURPDS means more or less that the subject of the purposive clause is different from both subject and object of the main clause.

(6.60)

Ngayu	nhawungarra-ma-rnuru	thamiini-ngu	pawulu-u
1.SG.NOM	look.after-CAUS-PRES	DASo+1POSS-GEN	child-ACC
	nguyirri-l	wanti-wala	
	asleep-THEN	lie-PURPDS	

I'm looking after my grandson's child so that he (grandson) can have a sleep.

The purposive suffix allows the lower subject to be omitted. Instead, it is identified with a matrix participant. The problem is that the matrix participants cannot be identified other than by their φ–features, for everything else has been discharged. There are basically two ways of getting around this problem. One will be sketched in the next section: there we shall reinterpret case assignment to account for non-configurationality. Here we will outline the second approch which consists in making the notion of a **pivot** explicit in syntax. Before we give the analysis of the Martuthunira data we shall motivate the introduction of pivots.

Take a look at coordination of finite clauses:

(6.61) The man got up and fell.

In this situation it is possible to use standard coordination rules: the verbs got up and fell can be coordinated with a missing subject and the missing subject is added later. The same can obviously be done in the following situation:

(6.62) The woman hit the man and fell.

Here, the phrases that are being coordinated are hit the man and fell. And in both cases, the woman is the subject. So far the facts are unproblematic. Yet, there are languages in which matters are less straightforward. Consider the following sentences from Dyirbal:

(6.63)

Ba-yi	yarra-∅	ba-rnngu-n	rdugumbi-rru	balga-n
DEIC-ABS.1	man-ABS	DEIC-ERG-2	WOMAN-ERG	hit-TNS
bardi-rnu				
fell down-TNS				

*The woman hit the man and he/*she fell down.*

	Ba-gu-l	yarra-gu	ba-la-n	rdugumbil-∅	balgal-nga-rnu
	DEIC-DAT	man-DAT	DEIC-ABS-2	woman-ERG	hit-ANTI-TNS
(6.64)	bardi-rnu				
	fell down-TNS				
	*The woman hit the man and she/*he fell down.*				

In Dyirbal, it is not possible to conjoin two verbs such that the subject of the two is the same if only one of them is transitive. To account for the difference between English and Dyirbal the notion of a **pivot** is introduced. (Other terminology is **privileged syntactic argument** (**PSA**). Although these two concepts are not the same, it is very difficult to discern actual differences.) The pivot is the central element that steers the linking from finite clauses. One assumes that each finite clause introduces a pivot. If two sentences are conjoined, then what gets shared is not the subject but the pivot. One says that Dyirbal has an S/U-pivot, meaning that the shared constituent must be either intransitive subject or transitive object (=undergoer). English on the other hand has S/A-pivot. One may also speak of absolutive versus nominative pivot. Dyirbal seems to offer itself to the same analysis as we gave for English: the pivot is identified by its case, and therefore we can invoke the ordinary coordination rule. The only difficulty is the nonconfigurationality of Dyirbal. The relevant constituent that is coordinated is not formed at the surface structure. Therefore this strategy is not open to us. To make matters worse, there are languages which have a S/A-pivot while case marking is ergative (see Dixon [28]). Hence, in these languages the case marking cannot be used to identify the pivot. Finally, the pivot may additionally depend on the nature of the clauses. So there is no obvious connection between the pivot and the case it bears; neither is there a fixed relation between the pivot and its grammatical relation. Yidiny is a language with split–ergative case marking. The pivot is however chosen exactly according to its case. If it is a pronoun, then it is a pivot if it is marked nominative and hence is intransitive subject or transitive subject, but if it is a full NP it is pivot if it has absolutive case, and hence is either an intransitive subject or a transitive object. Typically, a sentence only has one pivot, but if one analyses the data closer then it appears that there is also the notion of a **secondary pivot**. Typically, the subject and the object divide among themselves the roles of pivot and secondary pivot. Either the subject is pivot and then the object is a seocndary pivot, or else the object is pivot and the subject is secondary pivot. However, we shall assume that intransitive clauses can have secondary pivots, and that transitive clauses can have secondary pivots that are not direct objects or subjects. Such will be the case with Martuthunira.

In order to account for the notion of a pivot, we shall divide the φ–stratum into two substrata. The first carries the hitherto established features and the second, call it the π–**stratum** carries information concerning the privilege of the referent. There is a feature π with values + or −. Thus, entries can have the following shape:

(6.65) $\langle x : \flat : [\pi : +]\rangle, \quad \langle x : \flat : [\pi : -]\rangle, \quad \langle x : -\rangle$

where $\flat$ is either $\vartriangle$, $\triangledown$ or $\lozenge$. In the first case we call x the **privileged referent**, in the second case the **secondary privileged referent**. The privileged referent will play the role of the pivot, the secondardy privileged referent the role of the secondardy pivot. Notice that it would not have been enough to just install a new feature into the φ–stratum to encode the privilege. For suppose that there are two referents, one with masculine gender and one with feminine gender. Then it would have been possible for both to have the feature $\pi : +$ and hence to be privileged, while this would not have been possible with two referents carrying the same gender. But the number of privileged referents does not depend on their gender. There can always only be one such referent.

With this adjustment being made, we must return to the agreement system. What we shall assume is that when agreement is attached to the verb, the following mechanism is activated:

1. The grammatical relation is removed.

2. The case assignment is added at the case stratum.

3. The φ–features are added at the φ–stratum.

4. If the argument is the pivot, then the referent is made the privileged referent, if the argument is secondary pivot then the referent will be made the secondary privileged referent.

Of course, it must be stated explicitly how to assign pivot and secondary pivot. As we have outlined above, this is a subtle matter, involving many factors. Here is the argument structure for agreement object for an accusative language, with

S/A-pivot and the accusative object to the right of the verb (eg English).

(6.66)
$$\begin{array}{|l|}\multicolumn{1}{c}{/\textsc{agro}/}\\\hline \langle e : \lozenge\oslash : [\textsc{trs} : +]\rangle \\ \langle x : \triangledown\ominus : [\textsc{case} : acc]\rangle \\\hline \langle x : \lozenge : [\quad]\rangle \\\hline \langle x : \lozenge : [\pi : -]\rangle \\\hline \langle x : \triangledown : [\textsc{gr} : \mathit{2}] \\\hline \varnothing \\\hline \varnothing \\\hline \end{array}$$

Hence, a fully inflected verb shows also the privilege of its referents. This depends or may depend on the nature of the clause that is being projected. Thus, the agreement suffix is sensitive to the type of clause, and may either make the object or the subject the pivot, depending on the type of clause. Infinitives do not trigger agreement, but they do project a pivot. In fact, we shall assume that each clause contains a pivot, and a secondary pivot if it is transitive. In this way we can account for a number of facts.

With the notion of pivot and secondary pivot introduced we shall proceed to an analysis of the Martuthunira purposive clauses. Now, Martuthnuira is an accusative language, hence has S/A–pivots (it is not known that any language which is accusative has S/U–pivot, while there are ergative languages with S/A–pivot). As the main clause signals what is pivot and what is secondary pivot, and since pivot is subject, we may assume that PURPSS identifies the lower subject with the main pivot:

(6.67)
$$\begin{array}{|l|}\multicolumn{1}{c}{/\textsc{purpss}/}\\\hline \langle e : \lozenge\oslash : [\quad]\rangle, \\ \langle e' : \triangledown\oslash : [\textsc{purp} : +]\rangle \\\hline \langle y : \lozenge : [\pi : +]\rangle \\\hline \varnothing \\\hline \langle x : \triangledown : [\textsc{gr} : \mathit{1}]\rangle \\\hline \varnothing \\\hline e' \\\hline y \cap x \neq \varnothing; \\ \mathsf{purp}'(e) \doteq e'. \\\hline \end{array}$$

The semantics of this item is relatively complex. It states that when the suffix is added to the verb (here represented by e') then the GR ‘1’ is removed from x, while the verb picks up a pivot y, and states that x and y are not disjoint. Furthermore, e is done for the purpose of e'. Similar analyses can be given for the other markers. Crucially, the object agreement marker PURPS=O makes use of the secondary pivot. Here the assumption comes into play that arguments other than the direct object can be secondary pivots. In Martuthunira, any accusative complement (direct object, indirect object and benefactive) can be a pivot. So, we give the following analysis for PURPS=O:

(6.68)

<table>
<tr><td>/PURPS=O/</td></tr>
<tr><td>$\langle e : \Diamond\oslash : [\ \ ]\rangle$,
$\langle e' : \triangledown\oslash : [\text{PURP} : +]\rangle$</td></tr>
<tr><td>$\langle y : \Diamond : [\pi : -]\rangle$</td></tr>
<tr><td>$\varnothing$</td></tr>
<tr><td>$\langle x : \triangledown : [\text{GR} : \mathit{1}]\rangle$</td></tr>
<tr><td>$\varnothing$</td></tr>
<tr><td>e'</td></tr>
<tr><td>$y \cap x \doteq\!\!\!\!/\ \varnothing$;
$\mathsf{purp}'(e) \doteq e'$.</td></tr>
</table>

We have earlier spoken of the fact that no two different accusative arguments can head a separate purposive clause. Now, this receives an explanation from the fact that although there can be several accusative arguments, there can be only one secondary pivot, even though the choice of secondary pivot is free.

Various things follow from this analysis:

1. The purposive suffix is added after the grammatical roles have been added.

2. The purposive suffix is added before case assignment takes place. Hence no subject agreement is triggered.

While the first is true, the second will be seen to be only partially correct. Since there is no verbal agreement, one cannot see whether subject agreement is triggered, and we shall see that subordinate clauses in certain languages sometimes have both reference tracking markers and agreement markers. Furthermore, there do exist sentences in which both the main clause subject and the lower clause subject are present, whence it is actually too much to require that the grammatical relation is deleted. Here is an example:

(6.69)

`Nganaju`	`mimi`	`warrirti-i`	`panyu-ma-lalha,`	`ngaliya`
1.SG.GEN	uncle	spear-ACC	good-CAUS-PAST	1.DU.EXCL
`puni-lu`	`murla-a`	`manku-lu`		
go-PURPSS	meat-ACC	get-PURPSS		

My uncle fixed a spear so that we two could go to get meat.

Notice that if the subject of the purposive clause would not be overt it would have to interpreted that the uncle was going alone. Hence, we assume that the case marking is retained, and that the subject of the purposive clauses can be omitted only in case strict coreferentiality obtains.

Now let us briefly return to Dyirbal. We shall first of all remark that Dyirbal seems to have simple coordination.

(6.70)

`nguma`	`yabu-nggu`		`ngamba-n`
father-ABS	mother-ERG		heard
`nguma`	`yabu-nngu`	`bura-n`	`ngamba-n`
father-ABS	mother-ERG		saw and heard

(6.71)

`nyurra`	`ngana-na`	`ngamba-n`
you all	us	heard
`nyurra`	`ngana-na`	`bura-n ngamba-n`
you all	us	saw and heard

Here, the subjects are identified and the objects. This would however be expected on either analysis: the objects are both the privileged actants and the subject the secondary privileged actants. Hence an alternative analysis, where the pivots and the secondary pivots are identified is also plausible. However, the latter analysis runs into difficulties with the case assignment. This is already the case when we consider only the identification of the pivots. For we have stated above that in certain languages the cases of the pivots may be different depending on the clause type, so that standard coordination cannot be applied here.

We shall close this section with examples from Plains Cree, an Algonquian language. This language does not mark argument relations by means of cases but by using the primary and secondary pivot. Our discussion is based on [94] (though the data is from [103]). This will also shed light on the notion of obviation, which appeared in Potawatomi (see 3.5). In Plains Cree the verbal arguments are ranked as follows:

2 > 1 > 1.DU.INC > 3.PROX > 3.OBV

Suppose that a verb has two arguments. To tell which is subject and which is object the following convention is employed.

1. If the verb has the suffix glossed as DCT then an argument of the verb is subject if it is higher on this scale than the other one.
2. If the verb carries the suffix glossed as INV then the higher ranked argument is the object.

This takes care of the case when they have different rank. We can only only speculate for lack of data what will happen when they have equal rank. In this case, if we take 1st and 2nd person, the verb is reflexive and will most likely be marked by a reflexive marker and hence count as intransitive. Moreover, since the referent is usually uniquely fixed in this case, there is little danger of confusion. By applying the above rules, the following examples are now readily understood:

(6.72) `Ki-wāpam-i-n`.
2.SG-see-DCT-1.SG
You see me.

(6.73) `Ki-wāpam-iti-n`.
2.SG-see-INV-1.SG
I see you.

Now we turn to third person arguments. They fall into two classes, namely **proximate** and **obviative** (glossed as PROX and OBV). If only one 3rd person argument appears, it may either be proximate or obviative. If two 3rd person arguments cooccur in a sentence, then one must be proximate and the other obviative. The following examples demonstrate this system.

(6.74)

`Wāpam-ē-w`	`nāpēw-∅`	`atim-wa`
see-DCT-3SG.PROX	man-PROX	dog-OBV

The man sees the dog.

(6.75)

`Wāpam-ik`	`nāpēw-∅`	`atim-wa`
see-INV-3SG.PROX	man-PROX	dog-OBV

The dog sees the man.

(6.76)

Wāpam-ik	nāpēw-a	atim-∅
see-INV-3SG.PROX	man-OBV	dog-PROX

The man sees the dog.

(6.77)

Wāpam-ē-w	nāpēw-a	atim-∅
see-DCT-3SG.PROX	man-OBV	dog-PROX

The man sees the dog.

We can account for these facts as follows. We assume that the semantics of DCT and INV is as outlined above. (It is a little bit awkward to write down the semantic structures, since they consist in a disjunction.) Namely, INV is put if the subject is 2nd person, or the subject is 1st person and the object is not 2nd person etc. However, if one regards these markers as a fusion of subject and object agreement, then there are not so many cases at all to consider in comparison to other languages. The interesting part is now the 3rd person arguments. We shall assume that any argument can be either pivot or secondary pivot, but while 1st and 2nd person go unmarked, a 3rd pivot is marked PROX and a 3rd person secondary pivot is marked OBV. The agreement rule in this case is that DCT is put if the subject is pivot and the object is secondary pivot, and otherwise INV is added. It is clear that there can never be more than one pivot, and so there is no problem in assigning the correct relation to the arguments.

It may be argued against this solution that it is not at all clear that the proximate/obviative distinction relates to the notion of pivot in Plains Cree. If that is so, this does not speak against the present analysis, rather only against the use of the term 'pivot'. It does appear however that the two are related. [5]

6.7 Anaphora

In this section we shall discuss a phenomenon that has attracted a lot of attention in the last decades: anaphora. In view of the many facets that have been discovered, our aim can only be a modest one. We shall show that our model provides a tool for establishing referents for anaphora. This is distinct from DRT in the sense that DRT does not talk about how an anaphor is actually linked to a referent; rather, DRT is concerned with conditions under which a given linking yields a well–defined structure. For example, DRT explain that if a referent is introduced

[5] According to [94], 1st and 2nd are always -OBV.

inside an implication or a negation then it is inaccessible for reference outside the implication or negation, respectively. Thus the ungrammaticality of the following sentences is accounted for under the obvious linking:

(6.52) If Susie saw [a unicorn]$_1$, she was happy. *It$_1$ was beautiful.
(6.53) Susie didn't see [a unicorn]$_1$. *It$_1$ was beautiful.

So, DRT explains that if it is linked to the referent of the unicorn in the DRS then the discourses (6.52) and (6.53) are infelicitous. However. DRT does not tell us why these discourses are infelicitous, since it does not explain why we should opt to relate the anaphor it to the referent of the unicorn. This latter fact is however what we shall try to explain here.

We shall start with pronouns. The basic intuition is that a pronouns picks up a referent by means of its φ–features. Since the φ–features are present even after the completion of the sentence, this is in fact possible. So, let us assume the following lexical entry for pronouns:

$$\begin{array}{|l|}
\multicolumn{1}{c}{\text{/it/}} \\
\hline
\langle x : \vartriangle \ \varnothing : [\textsc{case} : \mathit{nom}] \rangle \\
\hline
\langle y : \lozenge : \left[\begin{array}{lcl} \textsc{pers} & : & \mathit{3} \\ \textsc{class} & : & \mathit{neut} \\ \textsc{num} & : & \mathit{sg} \end{array}\right] \rangle \\
\hline
\varnothing \\
\hline
\varnothing \\
\hline
\varnothing \\
\hline
x \\
\hline
x \doteq y \\
\hline
\end{array}$$

It is obvious how the entries for the other pronouns are. Notice first of all that the features for the pronouns are distributed over x and y. In fact, a fuller representation would be as follows:

/it/

$$
\begin{array}{|l|}
\hline
\langle x : \triangle\ \ \varnothing : [\textsc{case} : nom] \rangle \\
\hline
\langle y : \Diamond : \left[\begin{array}{lcl} \text{PERS} & : & \mathit{3} \\ \text{CLASS} & : & neut \\ \text{NUM} & : & sg \end{array} \right] \rangle \\
\langle x : \Diamond : \left[\begin{array}{lcl} \text{PERS} & : & \mathit{3} \\ \text{CLASS} & : & neut \\ \text{NUM} & : & sg \end{array} \right] \rangle \\
\hline
\varnothing \\
\hline
\hline
\varnothing \\
\hline
\varnothing \\
\hline
\hline
x \\
\hline
x \doteq y \\
\hline
\end{array}
$$

However, we shall work with the tacit assumption that identical referents share the same φ–features by default. This is so since φ–features are connected with elements in the model, and if two referents are connected to the same element they shall have the same φ–features. (This is actually not quite correct. We shall return to this issue below in the last section.) Now, we shall assume that if a pronoun is consumed by a verb, it cannot identify both x and y at the same time. [6] So, when the sentence is complete, what is left of a pronoun is the following

[6]The reader is made aware of the fact that we have left it unclear which are the precise conditions under which a referent which does not belong to a case–stratum is identified. Obviously, the details of this process must be clarified. The optimal scenario would be one in which there is uniform rule, say that the maximum number of referents can be identified. However, it seems to us that such a simplified view is untenable. We must therefore assume that it is possible to dictate that certain referents cannot be identified under certain conditions while others must be.

argument structure

$$
\begin{array}{c}
\text{/it/} \\
\begin{array}{|l|}
\hline
\varnothing \\
\hline
\langle y : \Diamond : \left[\begin{array}{lcl} \textsc{pers} & : & \mathit{3} \\ \textsc{class} & : & \mathit{neut} \\ \textsc{num} & : & \mathit{sg} \end{array}\right]\rangle \\
\hline
\varnothing \\
\hline
\hline
\varnothing \\
\hline
\varnothing \\
\hline
\hline
y \\
\hline
\delta(y) \\
\hline
\end{array}
\end{array}
$$

Hence, the referent y is open for identification under merge. Now notice that the argument structures of the first sentences of (6.52) and (6.53) have the following form:

$$
\begin{array}{c}
\text{/Susie didn't see a unicorn/} \\
\begin{array}{|l|}
\hline
\varnothing \\
\hline
\langle x : \triangle : \left[\begin{array}{lcl} \textsc{pers} & : & \mathit{3} \\ \textsc{class} & : & \mathit{fem} \\ \textsc{num} & : & \mathit{sg} \end{array}\right]\rangle \\
\langle y : \triangle : \left[\begin{array}{lcl} \textsc{pers} & : & \mathit{3} \\ \textsc{class} & : & \mathit{neut} \\ \textsc{num} & : & \mathit{sg} \end{array}\right]\rangle \\
\hline
\langle x : \triangle : [\pi \ : \ +]\rangle \\
\hline
\hline
\varnothing \\
\hline
\varnothing \\
\hline
\hline
x \\
\hline
\eta(x, y) \\
\hline
\end{array}
\end{array}
$$

The difference between these two is merely that in (6.52) $\eta(x, y)$ is an implication of the form $[y : \eta_1]$:$[\varnothing : \eta_2]$ while in (6.53) it is a negation of the form $\neg\ [y : \eta_1]$. In both cases the referent y is embedded in such a way that it is inaccessible from the main box. Assume for simplicity that intersentential coordination takes two struc-

Figure 6.8: Illicit Reference

/(6.53)/

$\varnothing$

$$\langle y : \triangle : \begin{bmatrix} \text{PERS} & : & \mathit{3} \\ \text{CLASS} & : & \mathit{neut} \\ \text{NUM} & : & \mathit{sg} \end{bmatrix} \rangle$$

$$\langle x : \triangle : \begin{bmatrix} \text{PERS} & : & \mathit{3} \\ \text{CLASS} & : & \mathit{neut} \\ \text{NUM} & : & \mathit{sg} \end{bmatrix} \rangle$$

$\langle x : \triangle : [\pi \ : \ +] \rangle$

$\varnothing$

$\varnothing$

z

$\neg$ | y, e |
|---|
| $\mathsf{see}'(e; y); \mathsf{unicorn}'(y).$ |

$\mathsf{susie} \doteq z; y \doteq x; \mathsf{beautiful}'(x).$

tures and merges as many referents on the φ–level as it can. Moreover, assume that it merges the DRS at main level. This then explains why the two discourses are infelicitous: the merging algorithm dictates that y of the first sentence is the antecedent of it, since they agree in their φ–features. However, this means that the referent for it occurs unbound in the DRS, and therefore the resulting DRS is not well–formed. The representation of (6.53) is shown in Figure 6.8 (certain details are suppressed). If on the other hand it is replaced by she the discourses are well–formed, since she will be referring to z and z is on the main DRS level.

Even if this model is overly simplistic it already has a great explanatory power. It supplements DRT with a necessary algorithm to establish the antecedent of a pronoun. It shows that the principles of DRT and the linking algorithm together are enough to explain in these basic examples why certain discourses are well–formed and others are not. Before we move on to consider more complex facts, we shall talk more explicitly about the background assumptions upon which the present proposal rests. First of all, we must assume that the default for φ–features is identification from the left. So, if a referent is a head at the φ–stratum then it must look for its argument on the left hand side. This will take care of the order sensitivity in pronominal reference as exemplified in the discourses below.

(6.54) Susie$_1$ was upset. Her$_1$ wallet was stolen.
(6.55) ?She$_1$ was upset. Susie$_1$'s wallet was stolen.
(6.56) ?She$_1$ was upset. Her$_1$ wallet was stolen.

The pronoun she cannot pick up a referent that is not yet introduced into the discourse. Or to be precise, in (6.55) the referent for Susie and for the pronoun cannot be identified since the first must be the argument, but the pronoun cannot take its argument to the right. In (6.54) the order is reversed and the discourse is fine. (6.56) is out since the first pronoun has no antecedent. Although we encounter frequently the situation where a pronoun opens a discourse, this must always be understood as elliptical. The antecedent must be interpolated. Hence we will rule out such situations.

Next, we shall turn to the sentence coordination. Recall that we have already introduced coordinators in Section 2.9. The principal device for handling coordination was the introduction of variables for argument structures. Thus it was possible to state an explicit restriction that any two constituents conjoined by and must have identical argument structures. Now that we have spread the argument structure into different strata, we must refine this analysis somewhat. It must be asked whether the identity condition is operative on all strata or whether there are some exceptions to it. We shall claim that the identity condition is operative on the case–stratum, the GR–stratum and the θ–stratum, while on the φ–stratum matters are different. [7] Basically, the φ–argument structures are free with respect to each other. Any two φ–argument structures are allowed to be combined. Moreover, pronouns can be resolved across constituents. This is shown in the following example.

(6.57) Susie talked to a student and gave him some advice.

Here, two verb phrases missing a subject are coordinated. However, in the second conjunct, a pronoun is put referring to an antecedent in the first conjunct. Notice

[7] The π–stratum seems to be an intermediate case. Roughly, it is preferred if the two conjuncts have the same π–stratum. This is an instance of syntactic parallelism. However, it does not always seem to be neceessary.

that the antecedent referent is an indefinite, so we predict that it would not be accessible if in place of and we put or. It is predicted therefore that the following discourse is infelicitous:

(6.58) ?Susie talked to a student or gave him some advice.

This discourse is however not as bad as (6.52) or (6.53). However, this may be because the referent is accommodated at the main level. It gets worse if we replace a by some. Moreover, the following examples are clear:

(6.59) ?Susie did not talk to a student and/or gave him some advice.

Hence we conclude that the same principles are operative with respect to φ–features as with sentential coordination. Thus, sentence coordination differs from and in that it coordinates main sentences. Hence, there are certain restrictions on the type of argument structures that it can coordinate, but other than that it functions in the same way as and. Whatever pronoun appears on the right hand constituent can be resolved by an object to the left, but if a pronoun appears on the left hand side it cannot be so resolved.

(6.60) ?Susie$_1$ talked to her$_2$ and gave Karen$_2$ her$_1$ advice.
(6.61) Susie$_1$ talked to Karen$_2$ and gave her$_2$ her$_1$ advice.

Now, which are the conditions of merge of the coordinators? The easier parts are the conditions on case, grammatical relations and θ–roles. Here we require identity (up to renaming, see Section 2.6). Therefore, one cannot compose an infinitive with a finite verb since the infinite has a 1 on the GR–stratum while the finite verb assigns nominative case but has no 1. The reasons for these restrictions are that each argument of either constituent is interpreted as belonging to both.

6.8 Disagreement

In this section we will reconsider the mechanics of φ–features in view of an unexpected phenomenon, namely that of *disagreement*. We will see below that disagreement is not so uncommon even in languages that seem to adhere quite strictly to agreement. The system developed so far seems to exclude the possibility of disagreement altogether. If two constituents disagree in their φ–features they cannot merge. However, matters are not that simple. Notice that there are two systems of φ–features in a language: a *formal* system of gender, person and number names, and a *semantic* system of gender, person and number classification. It is easiest to see the difference with gender. All diminutives in German are neuter, for example, das Bäumchen *the little tree* (but der Baum, which is masculine), das Bömbchen *the little bomb* (but die Bombe, which is feminine). This is a purely formal system, witness the fact that there is the noun das Mädchen (the girl). This noun is in fact not even a diminutive noun in the present day language. The semantic intuitions are quite clear: non–animate things are neuter, animate beings fall into either masculine or feminine. Therefore, das Mädchen is formally neuter but semantically feminine. Now consider the following texts.

(6.62a) The girl in the first row is very clever. She solves all problems in a few seconds.

(6.62b) Das Mädchen in der ersten Reihe ist sehr klug. Es/*Sie löst alle Aufgaben in wenigen Sekunden.

If a pronoun is sensitive to the semantic features, we should expect that the pronoun in the second sentence is feminine, since we are talking about a girl. In English this is borne out ((6.62a)). Agreement with respect to formal features seem also possible. In German, however, we only get neuter agreement! This is quite in line with the formal setup. For we have made pronouns sensitive to the formal features and not the semantic ones. Hence the German data supports the view that pronouns choose their antecedent using the formal features. In English this is not so not even in subordinate clauses.

(6.63a) [The girl in the first row]$_1$ believes that she$_1$/*it$_1$ is clever.

(6.63b) [Das Mädchen in der ersten Reihe]$_1$ glaubt daß es$_1$/*sie$_1$ klug ist.

However, it cannot be said that pronouns are either sensitive to formal gender or just to semantic gender or perhaps both; it also depends on the type of construction that is involved. Notice that pivot change is likewise sensitive to these factors, so there might be a deeper connection here.

It is strictly speaking not necessary to speak of formal and semantic gender. We might namely say that when the formal feature is defunct it gives rise to a semantic property which simply classifies the object in just the same way. So, we introduce new predicates, s-fem, s-masc and s-neut, which are true of objects just in case they are syntactically feminine, masculine or neuter. This is a daring proposal in view of the fact that it predicts that no matter how we refer to an object, the description will always have the same syntactic gender. It is possible to refute that using German data, and this would show that such argumentation would run into difficulty. (For example, suppose that the girl in the first row is called Sofie. Now replace the expression das Mädchen in der ersten Reihe by

Sofie. Now you must use feminine agreement.) But we will give a somewhat more spectacular case from Serbo–Croat, cited here after Corbett [21].

(6.64) ovi privatne zanatlije
these.MASC.PL private.FEM.PL artisan.PL
these private artisans

What we find is that the gender switches from feminine to masculine. There are some words which allow in the plural both modifiers, masculine and feminine. Therefore, it is legitimate to modify zantlije by a feminine attribute. By the rules of formal gender features, this means that throughout the whole DP everything must show feminine agreement. But in this example the determiner shows masculine agreement. Corbett notes that the shift is not random. He states the following rule.

> If stacked targets show different agreement forms, the target further from the controller will show the form with greater semantic justification.

We have shown decisively that the formal and the semantic features coexist and that they diverge. Moreover, we have to explain why it is that the semantic gender wins in the long run. To see that we have to unfold the story of agreement in φ–features once more. We have previously assumed that every morpheme has some meaning. So, if for example, an adjective is inflected for gender, then this gender will actually carry some meaning, and the same for number. This leads to an abundant iteration of some fact, such as that a given object is a plural entity or has feminine gender. But this could be tolerable was it not that it leads to incorrect predictions. One we have seen already: the syntactic gender does not need to bear any relationship with the semantic gender, and therefore the gender agreement morpheme for adjectives does nothing but transform the gender feature. It has absolutely no semantics. A similar fact can be noted with respect to plural. There are certain nouns which can only be used in the plural, and some nouns which are used in the plural but with singular meaning. An example is Latin Kalendae, which are plural feminine and denote the first day of the month. They trigger plural agreement on the adjective, but this time the meaning of the plural morpheme must be empty so as not to lead to a contradiction. Here is an English example.

(6.65) These scissors need to be sharpened.

This sentence can mean that you have only one pair of scissors, and they need to be sharpened. This is one object (arguably) and therefore the plural agreement on the verb would be contradictory if it would mean that the subject must be plural.

The question is therefore where if anywhere we shall put the semantic contribution of the φ–features? The idea is simple: if a lexical entry brings them with it, they need not be added. So, we only need to care once they can be added freely. In that case, they carry meaning just for those words that are arguments for the referent. In the case of a nominal referent, only the semantic feature of the noun can mean something. If a noun can be both singular and plural, then singular and plural mean what they typically mean. If there is no such choice, then they are vacuous. For example, nouns typically are specified for gender. Hence this gender will not make any semantic contribution throughout the sentence, even throughout the whole discourse, because no gender agreement morpheme that is introduced for agreement with that particular referent will make a semantic contribution. For example, in (6.62b) the discourse does nowhere imply that the girl is neuter, even though the pronoun has neuter gender. However, there are nouns that exist in both genders. In German, many professions have a masculine and a feminine form, and the latter is typically (but not necessarily) derived from the former by adding the suffix -in.

Masculine	Feminine	
Lehrer	Lehrerin	*teacher*
Professor	Professorin	*professor*
Bäcker	Bäckerin	*baker*
Pfleger	Schwester	*nurse*

The following solutions suggest themselves. We may assume that some nouns are not specified for gender. The masculine forms are produced by adding -∅, the feminine forms by adding -in. The second solution is to assume that the base form is actually masculine and that the suffix -in transforms the gender into feminine. We prefer the second solution because it explains why the masculine is the generic name. (Of course, this does not work for languages that fail to exhibit gender agreement. But then nothing needs to be done anyway.) Now, the crucial ingredient is the following. We assume that when the φ–feature is transformed then this transformation must reduce the tension between syntactic and semantic gender. In other words: the suffix -in, because it adds the meaning *feminine*, is allowed to transform the gender because it reduced the tension between semantic gender and syntactic gender. Hence, what we should not find is a suffix that makes, say, a noun denoting men but transforming the noun into a feminine one.

We conclude that language employs simultaneously a syntactic and a semantic gender system, like there are formal θ–roles and semantic θ–roles. The semantic gender system is more permanent than the syntactic one. Languages differ only in the persistence of syntactic or formal gender. We note with respect to German that the syntactic gender remains accessible even beyond the sentence boundary. Hence, German verbs do not throw away the syntactic referent of their arguments but only the case feature: the φ–features remain. We will claim that this is true for all languages. Hence, verbs only operate on the case features, but they do not modify any other features of the referent. In this way it is possible to use φ–features to refer to an antecedent across a sentence. However, we have seen that φ–features can also spontaneously change. We refer to this phenomenon as **spontaneous agreement change**. When it occurs, this signifies that the value the φ–feature changes into is actually semantically more accurate. This is the formal reflection of Corbett's observation. Languages differ in where spontaneous agreement change occurs. As a rule it can be noted that it is rare or happens only in a controlled way. Obviously, if it were rather liberal there would be hardly any way to track a discourse referent by means of its φ–features. In German, you can hardly get rid of the gender other than by introducing a new referent with different φ–features. This, however, is not to be confused with spontaneous agreement change.

(6.66a) Das Mädchen in der ersten Reihe ist Sofie. Es/Sie ist sehr intelligent.
(6.66b) *The girl in the first row is Sofie. It/She is very clever.*

There are also cases of spontaneous number and person change. Here is an English example of spontaneous number change.

(6.67) The fire–brigade came within five minutes. Soon they had everything under control.

One might have problems with this example since the firemen as a group are still different from a fire–brigade. Yet, under our extensional view there is actually no difference between these two groups. Moreover, it still needs to be explained why the plural pronoun happily finds an antecedent.

Spontaneous person change is actually pervasive. This is so since virtually all nouns are third person, while many of them can also be used as epithets or attributes of the speaker of hearer. In this case they must go along with a disagreeing controller. Prototypical examples are (6.65), (6.66) and (6.67). The English sentences are almost parallel cases to the German ones.

(6.68a) Ich, der König von Preußen, erkläre hiermit Euch, Bürger von Berlin, zu meinen Untertanen.
(6.68b) I, the king of Prussia, herewith declare you, people of Berlin, to be my subjects.
(6.69a) Lieber Gott, steh' mir armen Sünder bei!
(6.69b) Dear Lord, please help me poor sinner.
(6.70a) Ich, der Dir die ganze Zeit geholfen hat, muß mir nun Deine Klagen anhören.
(6.70a′) Ich, der ich Dir die ganze Zeit geholfen habe, muß mir nun Deine Klagen anhören.
(6.70b) I, who has helped you all this time, have to hear now all these complaints of yours.

Notice especially the two alternative sentences (6.70a) and (6.70a′). The verb in the relative clause of (6.70a) shows 3rd person agreement while it shows 1st person agreement in (6.70a′). This coincides with the occurrence of the 1st person pronoun in (6.70a′) and its absence in (6.70a). The relative clause in (6.70a) is actually syntactically complete, so the occurrence of the first person singular pronoun only helps to signal the agreement change.

We note therefore that especially the person agreement can change, but under very defined circumstances, namely mostly in postnominal modifiers. We shall finish off this section by a few remarks on the mechanics of spontaneous agreement change. It is clear that spontaneous agreement change, spontaneous though it may appear, is restricted to certain syntactic environments. Therefore, we assume that certain functional elements have the possibility to transform agreement. For example, in Serbo–Croat adjectives may transform the gender feature imported from the modified noun. In so doing, they contribute the meaning *is a group of women*. The case of appositions to personal pronouns is somewhat different. They are full DPs and therefore only tolerate postmodifiers. Generally, they agree with any postnominal modifier. However, it is neither possible to say that the pronoun changes the person spontaneously to its right (that would simply make wrong predictions as for verb agreement) nor is it possible to argue that the postnominal modifier transforms its person value, since there is simply no overt evidence for that. Instead, we assume that postnominal modifiers are actually underspecified for person. So, whatever functional element forms a postnominal modifier from a DP, it also makes its person feature maximally underspecified. This is in line with Corbett's observation only if the person feature of the personal pronoun to be modified is not third, because moving from a semantically incorrect gender to an unspecified gender is certainly going the right direction. Another possibility is to ascribe to the functional element the power to change the person feature. However, this approach is problematic since it allows the person feature to be changed without there being an overt element that signals this change.

(6.71a) Der dies schrieb, ist/*bin in großer Sorge um Dich.
(6.71b) *who this wrote, is/*am in deep worry about you.*

Finally, let us again look at the lexicon. We have said that elements may or may not bring their features instantiated from the lexicon. With a feature NUM that has three values, there are numerous possibilities as to what elements can exist in the lexicon. There is a popular theory, the theory of *blocking*, which would predict that if a more specific entry exists, the less specific one cannot. This seems to suggest that if we have an entry, say scissors then the entry scissor$^{\checkmark}$ will not exist. This is obviously true in this case, since there is no singular form available. But even if there is, still the two forms can coexist. An example is guts. It certainly exists in the lexicon with number unspecified. But the lexicon also contains the entry for guts, since its special meaning in the plural must be recorded. The reason why it does not block the form guts in the ordinary plural meaning of gut is precisely because the two mean different things.

Notes on this section. Pollard and Sag [80] discuss agreement and disagreement, particularly with respect to plural and gender at length. They reach the same conclusion, namely that neither a purely syntactic theory nor a purely semantic theory can deal with all the facts. The same is pointed out in Corbett [22]. However, they reject the view that φ–features originate in the noun and are copied onto the agreement targets, principally, because it introduces redundancy. But as our analysis shows, this conclusion is not warranted. First, language does need a certain redundancy (for example case agreement within a continuous noun phrase). It is probably a misconception to think that language always tries to be as efficient as possible. Second, agreement may be exploited for word order freedom and then become irredundant. Moreover, from a representational point of view, the redundancy does not appear in the referent systems because they store the φ–features of a given variable only once.

Index

Bibliography

[1] Dorith Abusch, *Sequence of tense and temporal* de re, Linguistics and Philosophy **20** (1997), 1 – 50.

[2] Judith L. Aissen and David M. Perlmutter, *Clause Reduction in Spanish*, Studies in Relational Grammar, Vol. 1 (David M. Perlmutter, ed.), The Chicago University Press, Chicago and London, 1983, pp. 360 – 403.

[3] Stephen R. Anderson, *A–morphous Morphology*, Cambridge University Press, Cambridge, 1992.

[4] ———, *Wackernagel's revenge: clitics, morphology, and the syntax of second position*, Language **69** (1993), 68 –98.

[5] Peter Austin, *Double Case Marking in Kanyara and Mantharta Languages, West Australia*, Double Case. Agreement by Suffixaufnahme (Frans Plank, ed.), Oxford University Press, 1995, pp. 363 – 379.

[6] Mark C. Baker, *Incorporation. A theory of grammatical function changing*, Chicago University Press, 1988.

[7] ———, *The polysynthesis parameter*, Oxford Studies in Comparative Syntax, Oxford University Press, 1996.

[8] Josef Bayer, *Comp in Bavarian Syntax*, The Linguistic Review **3** (1984), 209 – 274.

[9] Karl Bayer and Joseph Lindauer, *Lateinische Grammatik*, Oldenbourg, München, 1974.

[10] Tilman Becker, Owen Rambow, and Michael Niv, *The Derivational Generative Power of Formal Systems or: Scrambling is Beyond LCFRS*, Tech. report, The Institute For Research In Cognitive Science, University of Pennsylvania, 1992, IRCS Report 92–38.

[11] Sarah J. Bell, *Advancements and Ascensions in Cebuano*, Studies in Relational Grammar, Vol. 1 (David M. Perlmutter, ed.), The Chicago University Press, Chicago and London, 1983, pp. 143 – 218.

[12] Manfred Bierwisch, *On the grammar of local prepositions*, Syntax, Semantik und Lexikon, Studia Grammatica XXIX, Akademie Verlag, Berlin, 1988, pp. 1 – 65.

[13] Barry J. Blake, *Case*, Cambridge Textbooks in Linguistics, Cambridge University Press, 1994.

[14] Ol'ga Ju. Boguslavskaja, *Genitives and Adjectives as Attributes in Daghestanian*, Double Case. Agreement by Suffixaufnahme (Frans Plank, ed.), Oxford University Press, 1995, pp. 230 – 239.

[15] Denis Bouchard, *The Semantics of Syntax. A Minimalist Approach to Grammar*, The University of Chicago Press, Chicago, 1995.

[16] T. Burrow, *The Sanskrit Language*, Faber and Faber, London, 1973.

[17] Mike Calcagno, *A Sign–Based Extension to the Lambek–Calculuc for the Discontinuous Constituency*, Bulletin of the IGPL **3** (1995), 555 – 578.

[18] Bernhard Comrie, *Tense*, Cambridge Textbooks in Linguistics, Cambridge University Press, 1985.

[19] Bernhard Comrie, Maria Polinsky, and Ramazan Rajabov, *Tsezian languages*, The languages of the Caucasus, Curzon, London, 1999.

[20] Walter A. Cook S. J., *Case Grammar Theory*, Georgetown University Press, Washington, D. C., 1989.

[21] Greville Corbett, *Agreement: A Partial Specification Based on Slavonic Data*, Agreement in Natural Language. Approaches, Theories, Descriptions (Michael Barlow and Charles A. Ferguson, eds.), CSLI, Stanford, 1988, pp. 23 – 53.

[22] ———, *Gender*, Cambridge Textbooks in Linguistics, Cambridge University Press, Cambridge, 1991.

[23] David Crystal, *The Cambridge Encyclopedia of Language*, Cambridge University Press, Cambridge, 1987.

[24] William Davies, *Choctaw Verb Agreement and Universal Grammar*, Studies in Natural Language and Linguistic Theory, Kluwer, Dordrecht, 1986.

[25] Katherine A. Demuth, *Noun Classes and Agreement in Sesotho Acquisition*, Agreement in Natural Language. Approaches, Theories, Descriptions (Michael Barlow and Charles A. Ferguson, eds.), CSLI, Stanford, 1988, pp. 305 – 321.

[26] Alan Dench and Nicholas Evans, *Multiple case–marking in Australian languages*, Australian Linguistics **8** (1988), 1 – 47.

[27] Alan C. Dench, *Martuthunira. A Language of the Pilbara Region of Western Australia*, Pacific Linguistics, no. C – 125, Australian National University, 1995.

[28] R. M. W. Dixon, *Ergativity*, Cambridge Studies in Linguistics, no. 69, Cambridge University Press, Cambridge, 1994.

[29] Matthew S. Dryer, *Indirect Objects in Kinyarwanda Revisited*, Studies in Relational Grammar, Vol. 1 (David M. Perlmutter, ed.), The Chicago University Press, Chicago and London, 1983, pp. 129 – 140.

[30] Nicholas D. Evans, *A Grammar of Kayardild. With Historical–Comparative Notes on Tangkic*, Mouton de Gruyter, Berlin, 1995.

[31] Heinz Fähnrich, *Kurze Grammatik der georgischen Sprache*, Langenscheidt, Leipzig, 1993.

[32] Gisbert Fanselow, *Minimale Syntax*, Groninger Arbeiten zur germanistischen Linguistik, no. 32, Rijksuniversiteit Groningen, 1991.

[33] Donka Farkas and Henriëtte de Swart, *The semantics of incorporation: from argument structure to discourse transparency*, CSLI, Stanford, 2003.

[34] William A. Foley, *The Papuan Languages of New Guinea*, Cambridge Language Surveys, Cambridge University Press, Cambridge, 1986.

[35] Peter Geach, *A Program for Syntax*, Semantics for Natural Language (Donald Davidson and Gilbert Harman, eds.), Synthese Library, no. 40, Reidel, Dordrecht, 1972.

[36] Brendan Gillon, *Word order in classical sanscrit*, Indian Linguistics (1996).

[37] Brendan S. Gillon, *Word Order in Classical Sanskrit*, to appear in Indian Linguistics, 1995.

[38] Talmy Givón, *Syntax. A Functional–Typological Introduction*, vol. I, John Benjamins, Amsterdam, 1984.

[39] ______, *Syntax. A Functional–Typological Introduction*, vol. II, John Benjamins, Amsterdam, 1990.

[40] Jeroen Groenendijk and Martin Stokhof, *Dynamic Montague Grammar*, Papers from the Second Symposium on Logic and Language, Akademiai Kiadó, 1990, pp. 3 – 48.

[41] Claude Hagège, *La Structure des Langues*, Que–Sais–Je?, no. 2006, Presses Universitaires de Fance, Paris, 1982.

[42] Hubert Haider, *Deutsche Syntax – generativ. Vorstudien zur Theorie einer projektiven Grammatik*, Gunter Narr Verlag, Tübingen, 1993.

[43] Kenneth Hale, *Warlpiri and the grammar of nonconfigurational languages*, Natural Language and Linguistic Theory **1** (1983), 5 – 49.

[44] Morris Halle and Alec Marantz, *Distributed Morphology and the Pieces of Inflection*, The View from Building 20: Essays in Honour of Sylvain Bromberger (K. Hale and Keyser S. J., eds.), MIT Press, 1993, pp. 111 –176.

[45] Alice C. Harris, *Inversion as a Universal Rule of Grammar: Georgian Evidence*, Studies in Relational Grammar, Vol. 2 (David M. Perlmutter and Carol G. Rosen, eds.), The Chicago University Press, Chicago and London, 1984, pp. 259 – 291.

[46] Roparz Hemon, *Grammaire Bretonne*, 8 ed., Al Liamm, Rennes, 1975.

[47] Robert Hetzron, *Genitival Agreement in Awngi: Variation on an Afroasiatic Theme*, Double Case. Agreement by Suffixaufnahme (Frans Plank, ed.), Oxford University Press, 1995, pp. 325 – 335.

[48] Norbert Hornstein, *As time goes by. tense and universal grammar*, MIT Press, Cambridge, Mass., 1990.

[49] R. Huybregts, *Overlapping Dependencies in Dutch*, Utrecht Working Papers in Linguistics **1** (1984), 3 – 40.

[50] Hans Kamp and Uwe Reyle, *From discourse to logic, introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, Kluwer, Dordrecht, 1993.

[51] Ron Kaplan and Joan Bresnan, *Lexical–Functional Grammar: A Formal System for Grammatical Representation*, The Mental Representation of Grammatical Relations (Joan Bresnan, ed.), MIT–Press, Cambridge (Mass.), 1982, pp. 173 – 281.

[52] Fred Karlsson, *Finnische Grammatik*, Helmut Buske Verlag, Hamburg, 1984.

[53] Edward Keenan and Leonard Faltz, *Boolean semantics for natural language*, Reidel, Dordrecht, 1985.

[54] Edward L. Keenan, *On semantics and binding theory*, Explaining Language Universals (John A. Hawkins, ed.), 1988.

[55] László Keresztes, *Chrestomathia Mordvinica*, Tankönyvkiadó, Budapest, 1990.

[56] Tracy H. King, *SpecAgrP and Case: Evidence from Georgian*, MIT Working Papers in Linguistics **22** (1994), 91 – 110.

[57] August Klingenheben, *Deutsch–amharischer sprachführer*, Verlag Otto Harrassowitz, Wiesbaden, 1996.

[58] Jaklin Kornfilt, *Turkish*, Routledge, London and New York, 1997.

[59] Marcus Kracht, *On the Semantics of Locatives*, to appear in Linguistics and Philosophy, 1999.

[60] Paul Kroeger, *Phrase Structure and Grammatical Relations in Tagalog*, Dissertations in Linguistics, CSLI, Stanford, 1993.

[61] Fred Landmann, *Groups I*, Linguistics and Philosophy **12** (1989), 559 – 606.

[62] ______, *Groups II*, Linguistics and Philosophy **12** (1989), 723 – 744.

[63] Steven G. Lapointe, *Toward a Unified Theory of Agreement*, Agreement in Natural Language. Approaches, Theories, Descriptions (Michael Barlow and Charles A. Ferguson, eds.), CSLI, Stanford, 1988, pp. 67 – 87.

[64] Graham Mallinson and Barry Blake, *Language Typology*, North Holland, Amsterdam, 1981.

[65] Colin Masica, *The Indo–Aryan Languages*, Cambride Language Surveys, Cambridge University Press, 1991.

[66] P. H. Matthews, *Inflectional morphology. An introduction to the theory of word-structure*, Cambridge Textbooks in Linguistics, Cambridge University Press, 1978.

[67] R. S. McGregor, *Outline of Hindi Grammar*, Clarendon Press, Oxford, 1988.

[68] Glyn V. Morrill, *Type Logical Grammar. Categorial Logic of Signs*, Kluwer Academic Publishers, Dordrecht, 1994.

[69] Stefan Müller, *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche*, Linguistische Arbeiten, no. 394, Max Niemeyer Verlag, Tübingen, 1999.

[70] Klaus Peter Nickel, *Samisk grammatikk*, Universitetsforlaget, 1990.

[71] Rachel Nordlinger, *Constructive case. evidence from australian languages*, Dissertations in Linguistics, CSLI, Stanford, 1998.

[72] Jerry Norman, *Chinese*, Cambridge Language Surveys, Cambridge University Press, Cambridge, 1988.

[73] Toshiyuki Ogihara, *Tense, attitudes and scope*, Kluwer, Dordrecht, 1996.

[74] Terence Parsons, *Events in the Semantics of English. A Study in Subatomic Semantics*, Current Studies in Linguistics, no. 19, MIT Press, 1994.

[75] David M. Perlmutter, *Studies in relational grammar, vol. 1*, The Chicago University Press, Chicago and London, 1983.

[76] ______, *Studies in relational grammar, vol. 2*, The Chicago University Press, Chicago and London, 1984.

[77] David M. Perlmutter and Paul M. Postal, *Some Proposed Laws of Basic Clause Structure*, Studies in Relational Grammar, Vol. 1 (David M. Perlmutter, ed.), The Chicago University Press, Chicago and London, 1983, pp. 81 – 128.

[78] ______, *The 1–Avancement Exclusiveness Law*, Studies in Relational Grammar, Vol. 2 (David M. Perlmutter and Carol G. Rosen, eds.), The Chicago University Press, Chicago and London, 1984, pp. 81 – 125.

[79] Frans Plank, *(Re-)Introducing Suffixaufnahme*, Double Case. Agreement by Suffixaufnahme (Frans Plank, ed.), Oxford University Press, 1995, pp. 3 – 110.

[80] Carl Pollard and Ivan Sag, *Head–Driven Phrase Structure Grammar*, The University of Chicago Press, Chicago, 1994.

[81] Charles Randriamasimanana, *A study of causative constructions in malagasy*, Ph.D. thesis, University of Southern California, Los Angeles, 1981.

[82] Carol G. Rosen, *The Interface between Semantic Rules and Initial Grammatical Relations*, Studies in Relational Grammar, Vol. 2 (David M. Perlmutter and Carol G. Rosen, eds.), The Chicago University Press, Chicago and London, 1984, pp. 38 – 77.

[83] John Robert Ross, *Gapping and the Order of Constituents*, Progress in Linguistics (Manfred Bierwisch and M. Heidolph, eds.), Mouton, The Hague, 1970, pp. 249 – 259.

[84] Edward Sapir, *Language: an introduction to the study of speech*, Harcourt, Brace & World, New York, 1921.

[85] Philippe Schlenker, *A unified theory of pronouns, tense, and worlds in attitude contexts*, Lecture held at GLOW 99, 1999.

[86] Stuart Shieber, *Evidence against the Context–Freeness of Natural Languages*, Linguistics and Philosophy **8** (1985), 333 – 343.

[87] Anna Siewierska, *Person*, Cambridge Textbooks in Linguistics, Cambridge University Press, Cambridge, 2004.

[88] Andrew Spencer, *Morphological Theory. An Introduction to Word Structure in Generative Grammar*, Blackwell, Cambridge, 1991.

[89] Arnim von Stechow and Wolfgang Sternefeld, *Bausteine syntaktischen wissens*, Westdeutscher Verlag, Opladen, 1988.

[90] Mark Steedman, *Gapping as Constituent Coordination*, Linguistics and Philosophy **13** (1990), 207 – 263.

[91] ______, *Surface Structure and Interpretation*, Linguistic Inquiry Monographs, no. 30, MIT Press, 1996.

[92] Alan R. Thomas, *The Welsh Language*, The Celtic Languages (David MacAulay, ed.), Cambridge Language Surveys, Cambridge University Press, 1992.

[93] Miklós Törkenczy, *Hungarian Verbs and Essentials of Grammar*, Passport Books, Lincolnshire, Illinois, 1997.

[94] Robert D. Van Valin and J. LaPolla, Randy, *Syntax. Structure, meaning and function*, Cambridge Textbooks in Linguistics, Cambridge University Press, 1997.

[95] Kees F. M. Vermeulen, *Merging without Mystery or: Variables in Dynamic Semantics*, Journal of Philosophical Logic **24** (1995), 405 – 450.

[96] Kees F. M. Vermeulen and Albert Visser, *Dynamic bracketing and discourse representation*, Notre Dame Journal of Formal Logic **37** (1996), 321 – 365.

[97] Günther Webelhuth, *Principles and parameters of syntactic saturation*, Oxford Studies in Comparative Syntax, Oxford University Press, Oxford, 1992.

[98] Ilse Wegner, *Suffixaufnahme in Hurrian: Normal Cases and Special Cases*, Double Case. Agreement by Suffixaufnahme (Frans Plank, ed.), Oxford University Press, 1995, pp. 136 – 147.

[99] Chris Wilder, *Coordination, ATB and Ellipsis*, Minimalism and Kayne's Asymmetry Hypothesis (Jan-Wouter Zwart, ed.), Groninger Arbeiten zur Germanistischen Linguistik, no. 37, Groningen, 1994, pp. 291 – 329.

[100] ______, *Some Properties of Ellipsis in Coordination*, Tech. Report 2, Department of Linguistics, University of Geneva, 1995.

[101] Gernot Wilhelm, *Suffixaufnahme in Hurrian and Urartian*, Double Case. Agreement by Suffixaufnahme (Frans Plank, ed.), Oxford University Press, 1995, pp. 113 – 135.

[102] Arnett Wilkes and Nicholias Nkosi, *Zulu. a Complete Course for Beginners*, Teach Yourself Books, 1995.

[103] H. C. Wolfart, *Plains Cree: a grammatical study*, American Philosophical Society, Philadelphia, 1973.

[104] Jean Zemb, *Vergleichende Grammatik Französisch–Deutsch. Teil I: Comparaison de deux systèmes*, Bibliographisches Institut, Mannheim, 1978.