SYNTAX AND SEMANTICS: AN OVERVIEW
ARNIM VON STECHOW

1. Introduction................................................................................................................ ......... 3
2. A Fregian Conception of Syntax/Semantics ............................................................... 4
3. The Syntax/Semantics interface in Generative Grammar........................................... 7
   3.1. Generative conceptions of grammar ................................................................. 7
   3.2. Building Structures: External and Internal Merge ........................................... 9
   3.3. Notes on the literature....................................................................................... 12
4. A λ-language and the Interpretation of External and Internal Merge....................... 12
   4.1. Logical Form..................................................................................................... 12
   4.2. Syntax and Semantics of EL............................................................................ 13
   4.3. Interpretations of External Merge................................................................... 15
   4.4. Interpretation of Internal Merge................................................................. 16
5. The Two Most Important Rules of Construal: FI and QR...................................... 17
   5.1. QR resolves type-clashes................................................................................. 17
   5.2. QR and Scope Ambiguities............................................................................. 20
   5.3. QR binds pronouns......................................................................................... 21
6. Relative Clause and Quantifying into XP: The Empty Pronouns WH and PRO........ 23
7. Intensional Contexts: The Type Language IL ....................................................... 26
   7.1. Intension and Extension................................................................................... 26
   7.2. Syntax and Semantics of IL............................................................................ 27
   7.3. Notes on the literature.................................................................................... 32
8. Modality and Reconstruction..................................................................................... 33
   8.1. de re/de dicto................................................................................................... 33
   8.2. Specific de dicto.............................................................................................. 35
   8.3. Head movement.............................................................................................. 37
   8.4. Notes on the literature:................................................................................... 38
9. Questions: wh-Movement, Scope Marking, Pied-Piping ........................................... 38
   9.1. Constituent Questions...................................................................................... 38
   9.2. whether-Questions........................................................................................ 39
   9.3. Notes on the literature.................................................................................... 44
10. Comparative and Superlative..................................................................................... 45
   10.1. Comparative: QR and Ellipsis Interpretation............................................... 45
1. **INTRODUCTION**

Syntax is a device for generating the expressions of language. Semantics is the device that interprets the expressions by assigning them meanings. This article presents the syntax/semantics interface for a generative grammar in the style of the GB-theory and later developments (PPT, MP). It does so in developing syntactic and semantic structures for a range of interesting constructions of English (and occasionally other languages). And it shows how the semantic structures are obtained from the syntactic ones by rules of construal. The semantic structures have a precise interpretation, which is indicated. The constructions considered are in most cases extremely simple and neglect complicated issues that have been discussed in the literature. But they are complicated enough to see what is needed for the syntax/semantics interface. And they give a concrete idea what the rules/principles of construal might look like. If we looked at other languages, different and more principles might be needed. I have confined myself to a sort of toy box English, because this is the language that most linguists understand and use for theoretical investigations. The sketch of a theory of interpretation is appropriate for GB-theory and perhaps the Minimalist Program. Later architectures like Phases are more problematic. The picture I am drawing is certainly too simple (and occasionally wrong). But it is complicated enough. In particular, it makes an

---

*I wish to thank Atle Grønn for correcting the many typos that were in the original version of this article.*
essential use of all sorts of LF-movements. There are theories like variable free categorial or combinatory grammars that try to do everything without movement. I am not discussing these alternatives. They would be the topic for a different article. I have said something about these issues in (von Stechow, 1991), but the theory has developed since.

2. A FREGIAN CONCEPTION OF SYNTAX/SEMANTICS

Most theoreticians doing formal semantics of natural language take the following principles attributed to the German philosopher Gottlob Frege as guiding lines:

1. Compositionality (“Frege Principle’’): The meaning of complex expression is a function of the meaning of its parts and their syntactic composition.

2. Context principle: Make the meaning of words (and parts) so that the sentence meanings come out correctly by compositionality.

3. Sentence meanings: The meaning of an unembedded sentence is a truth-value (an extension). The meaning of an embedded sentence is a proposition (in intension).

The precise spell-out of the principles is under debate. We will see that sentence meanings can be more complicated depending on the range of semantic phenomena to be considered. In some theories, sentence meanings are context change potentials (functions from propositions to propositions), in other theories sentence meanings are (Kaplanian) characters (functions from contexts to propositions) and so on. We will touch upon these issues in the development of the article.

Here is a first illustration of Frege’s guide-lines.

Consider the sentences:

(1) a. Bill likes Mary.
   b. Mary likes Bill.

The sentences are made of the same material, but they obviously mean something different. Thus it is not enough to know the meanings of the words Bill, likes and Mary to determine the sentence meaning. The computation must take into account the syntactic structure. To calculate the meaning of (1a), first compute the meaning of the VP likes Mary from the meaning of likes and Mary by an appropriate semantic function/operation. Then we go on and compute the meaning of the sentence by combining the meaning of the subject Bill with that of the VP by means of another (or perhaps the same) semantic operation. The calculation of the meaning of (1b) uses the same semantic operations for the calculation, but the input is different, and therefore it doesn’t come as a surprise that the result reached for
the sentence might be different. Suppose Bill likes Mary, but she doesn’t like him. In that case, the first sentence is true but the second one is false.

The meanings of expressions are conventionally represented by the use of double square brackets \([ 1 ]\). For the truth-value “true” we use the number 1, for the truth-value “false” we use the number 0. The meaning of the two sentences can then be described as:

\[
(2) \begin{align*}
& (a. \ [\ [\text{Bill likes Mary}] \ ] = 1 \text{ iff Bill likes Mary}. \\
& (b. \ [\ [\text{Mary likes Bill}] \ ] = 1 \text{ iff Mary likes Bill}.
\end{align*}
\]

An inspection of the example shows the usefulness of the context principle. We have a clear intuition of what the sentences mean. But what do the words mean? Everyone would say that Bill and Mary denote particular persons known to the user in the context of use. But what does likes denote? We may say “the attitude/state of liking”. But this description would give us no idea of how we could combine this meaning first with the object and than with the subject, following the syntax of the sentences, i.e. the structure \([sNP [VP V NP]]\).

According to the context principle the following function is a useful meaning for likes:

\[
(3) \ [\ [\ [V \text{likes}] \ ] \ ] = \lambda x. \lambda y. y \text{ likes } x^1
\]

We assume that the meaning of the VP is obtained by applying the meaning of the verb to the meaning of the object via functional application (FA). Similarly the meaning of S is computed by combining the meaning of the VP with the meaning of the subject via FA. Here is the calculation of (2b):

\[
\begin{align*}
\ [\ [\ [\text{VP} [V \text{likes} [\text{NP Bill}]]] \ ] \ ] &= \ [\ [\ [\text{VP} [V \text{likes}]] (\ [\ [\ [\text{NP Bill}]] \ ] \ ] \ ] \ ] \text{ by FA} \\
&= \ [\ [\ [\lambda x. \lambda y. y \text{ likes } x]] (\ [\ [\ [\text{NP Bill}]] \ ] \ ] \ ] \ ] \text{ meaning of likes} \\
&= \ [\ [\ [\lambda x. \lambda y. y \text{ likes } x]] (\text{Bill}) \ ] \ ] \text{ meaning of Bill} \\
&= \lambda y. y \text{ likes Bill} \text{ function conversion (“\text{\textlambda}-conversion”)} \\
\ [\ [\ [s \ [\text{NP Mary} [\text{VP} [V \text{likes} [\text{NP Bill}]]]]] \ ] \ ] &=
\end{align*}
\]

---

1 The method of splitting an n-place function into a sequence of one-place functions is due to (Schönfinkel, 1924). I write functions in the \(\lambda\)-notation with the conventions of (Heim and Kratzer, 1998: Ch. 3). Metalinguistic sentences such as \(y \text{ likes } x\) stand for truth-values.
An even more dramatic illustration of the context principle would involve functional words like not, or, and, if…then, every, some or modals like must or can. No rational person would claim that he has a pre-theoretical intuition of their meanings.

Let us illustrate the third principle of Frege’s: embedded sentences don’t express truth-values but propositions.² It is easy to convince oneself that embedded sentences cannot express truth-values in the general case. Consider the following two examples:

(4) a. Bill believes that he, Bill, likes Mary.
     b. Bill doesn’t believe that Mary likes him.

The complement clauses are synonymous with (1a) and (1b) respectively. Suppose that both sentences are true, i.e., their meaning is 1. Then Bill would be an irrational person because he would believe and not believe the same meaning. But the scenario does not imply any irrationality of the subject. Possible world semantics solves this problem by providing propositions as meanings. The propositions for the complements in question are the sets of worlds \([\lambda w.\text{Bill likes Mary in } w]\) and \([\lambda w.\text{Mary doesn’t like Bill in } w]\).³ The two propositions are different. So Bill might believe the first and disbelieve the second without being irrational. The semantics has to say, of course, how it is possible that sentences have different meanings in different contexts. The interpretation principle that achieves that will be introduced below. For the beginner the idea that (unembedded) sentences should denote truth-values is puzzling. In order to understand that we have to think of sentences as presenting facts or non-facts of the real world: if a sentence presents a fact, it is true and if it represents a non-fact, it is false.

Notes on the literature: Frege’s Principle is never stated explicitly in the above form.

² Historically, this attribution is not entirely correct. Frege calls embedded meanings senses (“Sinne”) as opposed to non-embedded meanings, which he calls references (“Bedeutungen”). There is no agreement on what the proper reconstruction of ‘sense’ should be. Following (Carnap, 1947), semanticists reconstruct senses as intensions. See below.

³ I don’t distinguish between sets and their characteristic functions. In set notation, the first set would be \(\{ w | \text{Bill likes Mary in } w\}\).
by Frege himself. The passage that is mostly quoted is (Frege, 1923: p. 23). The Context Principle is stated in (Frege, 1884). The sense/meaning dichotomy is introduced in (Frege, 1892).

3. THE SYNTAX/SEMANTICS INTERFACE IN GENERATIVE GRAMMAR

3.1. Generative conceptions of grammar

The toy grammar underlying the examples of the last section is so to speak an ideal realisation of Frege’s theory of language. The syntax of a language generates trees and each node has a semantic interpretation. The interpretations of the terminal trees are taken from the lexicon. The interpretation of a branching node is obtained by the semantic values of its daughters via functional application (\(\text{FA}\)). Grammars of this kind exhibit *surface compositionality*: each constituent has exactly one semantic interpretation that is obtained from the meaning of its direct components by an appropriate semantic operation. There are grammars that try to realise this conception, e.g. variable free categorial grammars (see (Szabolcsi, 1989), (Jacobson, 1999), (Steedman, 2000) among others).

The widely accepted architecture among linguists working in the Chomsky tradition is that of GB-theory and more recent developments building upon it; cf. (Chomsky, 1981); (Chomsky, 1992). In this theory syntax generates structures, which may be called *Syntactic Structures (proper)* (SSs).\(^4\) These structures are transformed into semantic representations called *Logical Forms* (LFs) on the one hand and *Phonetic Forms/Representations* (PFs) on the other hand. The rules that translate SSs into LFs are called *rules of construal*, those that transform SSs into PFs are called *PF-rules*. The LFs are interpreted by semantic operations in the style of the previous sections. Thus the LFs form a language that has a Fregean interpretation. The PFs are the input for phonological rules.

It is important to note that SSs might be quite far away both from PFs and from LFs though for some simple constructions, like the examples discussed, the three structures might coincide. Chomskyan syntax is therefore an abstract system that typically does not meet the slogan WYSIWYG (“What you see is what you get”). The article will provide some examples that speak for the soundness of this conception. Here is the organisation of the GB-model:

(5) The GB-model

\(^4\) The label SS reminds of *surface structure*. In the last fifteen years, the term *spell-out* is used instead.
The syntax proper (GEN) generates deep structure (DS), which encodes grammatical functions: everything is in ‘its’ position, i.e. movement hasn’t occurred yet. After application of movement rules (e.g. A-movement as in passives, A-bar-movement as in relative clause or interrogatives, head movement like the movement of the finite verb to COMP in German) we reach the level of surface structure (SS). SSs are translated into LFs by rules like Quantifier Raising (QR) (see below) and by deletion of uninterpretable material (Full Interpretation). They are translated into PFs by rules like clitization, Focus Interpretation and perhaps others. The model of the Minimalist Program is alike with the difference that DS is eliminated and SS is called Spell-Out (SO).

(6) The MP-model

\[
\begin{align*}
\text{GEN} & \\
\downarrow & \\
\text{DS} & \\
\downarrow & \\
\text{PF} & \leftarrow \text{SS} \Rightarrow \text{LF}
\end{align*}
\]

The innovation of the Phase-model is cyclic branching: there is not only one branching point SS or SO but one at every “phase”, vP or CP, substructures that occur in the recursive generation of a syntactic structure.

(7) The Phase-model

\[
\begin{align*}
\text{GEN} & \\
\downarrow & \\
\text{PF}_i & \leftarrow \text{phase}_i \Rightarrow \text{LF}_i \\
\downarrow & \\
\downarrow & \\
\downarrow & \\
\downarrow & \\
\downarrow & \\
\downarrow & \\
\text{PF}_n & \leftarrow \text{phase}_n \Rightarrow \text{LF}_n
\end{align*}
\]

Different architectures of grammar may differ by:

- The type of grammar used for generating syntactic structures (context-free grammars, categorial grammars, Tree Adjunction Grammars, Montague-Grammar, Transformational Grammars and many others)
- The type of grammars used for the LF generation (typed languages with abstraction and possibly other syncategorematic rules, categorial grammars with type shifters,
DRT and so on)

- The PF rules and phonological rules in general
- The interface organisation

Linguists working in the Chomskyian tradition usually assume that LFs are made of the same material as SSs, i.e. they are trees. An important difference between syntactic trees and semantic trees is that the nodes of the former carry syntactic categories (N, A, P, V, and possibly others) but the latter carry semantic categories, called logical types such as e, t, et, and so on.

The heart of the syntax/semantics interface consists of the rules of construal, which convert SSs into LFs. The most restrictive interface theory is one that does it without such rules. Syntactic and semantic structures coincide and each node is directly interpreted. Some categorial grammars try to realise this ideal. A less restrictive (but more realistic) framework derives LFs from SSs by rules of construal. The GB-theory assumes that we first generate a complete SS, which we then transform into an LF. The LF is interpreted semantically. A phase model is more restricted. It translates each phase into an LF. These LFs may not undergo any essential change in later steps of the generation. The interpretation should use the meaning of the previous phase plus some syntactic information stemming from the current LF.

3.2. **Building Structures: External and Internal Merge**

In this article we stick to the GB/MP-model. We assume that SSs can be generated by a lexicon, phrase structure rules (nowadays called External Merge), and movement rules (called Internal Merge). The LFs will be trees with logical types as categories. The language has variables for each type and certain syncategorematic symbols like the \(\lambda\)-operator.

The SSs of a language are generated by the rules External and Internal Merge on the basis of a lexicon consisting of lexical trees, i.e. words dominated by a syntactic category and a logical type. We will represent lexical trees by labelled brackets or by terminating context-free phrase structure rules. Phrase structure rules have the advantage that they can be used for writing many trees at once. For instance, the rules (8a) stand for the trees \([Ne \text{ Bill}]\) and \([Ne \text{ Mary}]\).

(8) Some lexical entries

a. \(Ne \rightarrow \text{Bill, Mary}\)
b. $N_{et} \rightarrow \text{boy, child, student}$

b. $V_{et} \rightarrow \text{sleeps, snores}$

c. $V_{e(t)} \rightarrow \text{likes, hates}$

d. $\text{Det}_{e(t)(e(t))} \rightarrow \text{every, a, no}$

e. $A_{et} \rightarrow \text{nice, obnoxious}$

The rule *External Merge* (EM) takes two trees $\alpha$ and $\beta$ and combines them to the new tree $[X \alpha \beta]$, where $X$ is either the syntactic category of $\alpha$ or that of $\beta$. The daughter that projects its category to the new tree is called *head*. Here is a derivation of a SS for the sentence *Bill hates every obnoxious child*. For the time being, we ignore the types for the generation.

1. $[A \text{obnoxious}]$ Lexicon
2. $[N \text{child}]$ Lexicon
3. $[N [A \text{obnoxious}] [N \text{child}]]$ EM(1,2)
4. $[\text{Det} \text{every}]$ Lexicon
5. $[\text{Det} [\text{Det} \text{every}] [N [A \text{obnoxious}] [N \text{child}]]]$ EM(4,3)
6. $[V \text{hates}]$ Lexicon
7. $[V [V \text{hates}] [\text{Det} [\text{Det} \text{every}] [N [A \text{obnoxious}] [N \text{child}]]]]$ EM(6,5)
8. $[N \text{Bill}]$ Lexicon
9. $[V [N \text{Bill}] [V [V \text{hates}] [\text{Det} [\text{Det} \text{every}] [N [A \text{obnoxious}] [N \text{child}]]]]]$ EM(8,7)

Following common practice, we will write $X'$ for intermediate projections, XP for complete phrases. A complete V projection is also written as S, and a V with one or two objects is written as VP. In other words, the last line will also be written as:

(9) $[S [N \text{Bill}] [V [V \text{hates}] [DP [\text{Det} \text{every}] [N [A \text{obnoxious}] [N \text{child}]]]]]]$

The bar-notation has no theoretical significance, however.

*Internal Merge* (IM) is used to generate structures that involve movement. Consider, e.g. the complement clause in *Bill wonders who Mary likes*. It has the following structure:

(10) $[[N \text{who}]_{t1} [C \text{C} [S \text{Mary} [V \text{likes} t1]]]]$

Suppose we have already generated the structure $[C \text{C} [S \text{Mary} [V \text{likes} [N \text{who}]]]]$ using the lexicon and EM. (The generation involves the lexicon entry C, i.e. a lexical tree dominating the empty string and the NP who.)

n. $[C \text{C} [S \text{Mary} [V \text{likes} [N \text{who}]]]]$ generated via Lexicon and EM
The next step of the generation is this:

\[ n+1. \quad [[\text{NP who}]_1 [C \cdot C [S \text{Mary} [VP \text{likes} [\text{NP who}]]]]] \quad \text{by Internal Merge} \]

*Internal Merge* takes a subtree from a tree and adjoins it to the tree. As the name indicates, the material adjoined is not taken from outside but from inside of the structure. The lower copy in the movement chain is called *trace* of the moved constituent, here *who*. The moved constituent is called *antecedent* (of the trace). The MP assumes that both constituents *who* are fully present in the structure. The lower is deleted at PF, i.e. not pronounced. Most semanticists assume that the lower copy is empty in the syntax already. Furthermore they assume co-indexing between the higher copy, the antecedent, and the trace. (The latter assumption is not shared by minimalism either, but it is important for the interpretation.) So a more convenient representation of step \( n+1 \) for semantic purposes is therefore:

\[ n+1. \quad [[\text{NP who}]_1 [C \cdot C [S \text{Mary} [VP \text{likes} [\text{NP t}_1]]]]] \]

where \( t_1 \) is the empty string with an index. Another instance of the movement rule is *head movement*, e.g. the movement of the finite verb to COMP in German matrix clauses. As an instance, consider the sentence:

(11) **Der Fritz mag jedes Kind**

the Fritz likes every child

A standard assumption in German syntax is that the sentence is generated by moving *mag* to C and subsequently moving Fritz to [Spec, CP]:

\[ n. \quad [CP [\text{mag} C] [[\text{der Fritz}] [[jedes Kind] \text{mag}]]] \]

\[ n+1. \quad [CP [\text{mag}_1 C] [[\text{der Fritz}] [[jedes Kind] t_1]]] \quad \text{IM} \]

\[ n+2. \quad [CP [\text{der Fritz}][\text{mag}_1 C][t_2 [[\text{jedes Kind} t_1]]]] \quad \text{IM} \]

It is obvious that the two rules Merge heavily overgenerate. We have to build in constraints that rule out undesirable (“ungrammatical”) structures. This is the real work to be achieved by syntax, which however is not the concern of this article. It is sufficient to have an idea of how syntactic structures of language come into life by simple principles. I should add that the SSs in recent generative syntax are more complicated than the ones assumed in this article. There are many additional functional projections used for case checking and other purposes.

In the next section we will see how this kind of syntax can be combined with semantic
interpretation.

3.3. Notes on the literature

The different interface organisations of Generative Grammars are outlined in (Chomsky, 1981), (Chomsky, 1995), (Chomsky, 2001). The claim that the following analyses belong to these frameworks is to be taken *cum grano salis*. Many details, even important ones, will differ from current syntactic research. The overall picture of the syntax/semantics interface is however consistent with the architecture of Generative Grammar, I believe. The first clear conception of the syntax semantics interface is presumably due to R. Montague. Montague said that syntactic rules have meanings. Each syntactic rule is associated with a particular semantic operation (typically FA), which puts the meanings of the arguments of the rule together. This is a direct implementation of the Frege Principle; see (Montague, 1970a, Montague, 1970b, Montague, 1970c). We will see that Montague’s conception can be applied in a straightforward way to the Minimalist Program: The standard interpretation of External Merge will be FA, and that of Internal Merge will be FA plus λ-abstraction.

4. A λ-language and the interpretation of external and internal merge

4.1. Logical Form

The majority of semanticists use a typed language for the representation of LFs. The structures of these languages are rather close to SSs and they have a simple and transparent interpretation. A typed language consists of trees whose nodes are labelled with *logical types*. A type represents the type of meaning a node denotes. LF languages may differ in the following way:

1. The types are different. Besides of the basic types e,t mentioned in the first chapter we may have types like s (world), i (time), l (place), v (event). Accordingly, the complex types might be different from language to language as well.

2. The types may encode different kinds of meaning.
   a. In an extensional language, types encode extensions unambiguously.
   b. In an intensional language, types ambiguously encode extensions and intensions.
   c. In a character language, types ambiguously encode extensions, intensions and characters.
   d. In a dynamic language, types encode context-change potentials.
e. There are typed languages that don’t make the distinction between extension and intension at all, e.g. Cresswell’s \(\lambda\)-categorial languages.

3. The trees of LF-languages may differ with respect to the types admitted at the branching nodes. Every language admits the splitting of a type into a functor type and an argument type. But other splits are possible depending on the semantic operations available for the interpretation.

4. A language may differ with respect to the syncategorematic rules and symbols they admit. Most LF languages have variables and \(\lambda\)-abstraction.

4.2. *Syntax and Semantics of EL*

We start with an extensional \(\lambda\)-categorial language. It is the system used in the first twelve chapters of (Heim and Kratzer, 1998) \([= \text{H&K}]\). We call this language *EL* (“Extensional Language”). The categories of the language are the following types.

\[
\begin{align*}
(12) & \quad \text{The type system for EL} \\
& \quad \text{The basic types are } e \text{ (“entities”)} \text{ and } t \text{ (“truth-values”). The complex types are} \\
& \quad \text{generated by the rule: if } a \text{ and } b \text{ are types, then } (ab) \text{ is a type.}
\end{align*}
\]

Types of form \((ab)\) are called *functional types*. We will mostly omit the outermost brackets. There are other notational conventions for functional types: H&K use \(<a,b>\), Montague uses \(b/a\), \(b//a\) or \(b///a\), (Kratzer, 1977) uses \(b:a\), and (Ajdukiewicz, 1935) uses \(ba\); there are still more variants. The basic types are written in different ways as well. Most categorial grammarians follow Ajdukiewicz and write \(n\) (“name”) for \(e\) and \(s\) (“sentence”) for \(t\). (Cresswell, 1973) uses \(1\) for \(e\) and \(0\) for a propositional type. He writes our functional types \((ab)\) as \(<b,a>\). So there is no uniformity in the literature and the reader has to be aware of the particular notation used by different authors. Our type system is a simplified version of the H&K-notation.

The *semantic domains* for these types are the following ones.

\[
\begin{align*}
(13) & \quad \text{The semantic domains for these types are the following ones.} \\
& \quad \text{a. } D_e = E = \text{the set of individuals} \\
& \quad \text{b. } D_t = \{1,0\} = \text{the set of truth-values} \\
& \quad \text{c. } D_{ab} = D_b^{Da} = \text{the set of (partial) functions from } D_a \text{ into } D_b.
\end{align*}
\]

The LF-language EL consists of a *lexicon*, *variables* for any type and the following structures:

\[
(14) \quad \text{Syntax of EL}
\]
1. If \( \alpha \) is a lexical tree of type \( a \), \( \alpha \) is a tree of type \( a \).  \( \text{(L)} \)

2. If \( x \) is a variable and \( a \) is a type, then \( [a \ x] \) is a tree of type \( a \).  \( \text{(Var)} \)

3. If \( \alpha \) is a tree of type \( ab \) and \( \beta \) is a tree of type \( a \), then \( [\alpha \alpha \beta] \) is a tree of type \( b \).  \( \text{(FA)} \)

4. If \( \alpha \) and \( \beta \) are trees of type \( et \), then \( [et \alpha \beta] \) is a tree of type \( et \).  \( \text{(PM)} \)

5. If \( \alpha \) is a tree of type \( b \) and \( x \) is a variable of type \( a \), \( [ab \ x \alpha] \) is a tree of type \( ab \). \( \lambda \)

The syntactic rules have names reminding of their semantic interpretation (\( L = \) lexicon, \( \text{Var} = \) variable rule, \( \text{FA} = \) functional application, \( \text{PM} = \) predicate modification, \( \lambda = \lambda - \) abstraction). H\&K use the notation \( [ab \ x \alpha] \) for our tree \( [ab \lambda x \alpha] \).

The lexicon consists of lexical trees that are labelled with types. We have entries such as \([e \ Mary]\) or \([et \ snores]\). To simplify the notation, we write these as \( \text{Mary}_e \) and \( \text{snores}_{et} \) respectively. The LF for the sentence \( \text{Mary snores} \) is the following tree:

\[
(15) \quad [t [e \ Mary] [et \ snores]]
\]

The interpretation of the language is given by interpreting the lexicon and by stating recursive interpretation principles for the syntactic structures. The lexicon is interpreted by a function \( F \) that maps each word \( \alpha \) of type \( a \) to a meaning in \( D_a \). Here is the interpretation for some lexical trees.

\[
(16) \quad \text{Some entries of the semantic lexicon}
\]

\[
F(\text{Mary}_e) = \text{Mary}
\]

\[
F(\text{snores}_{et}) = \lambda x. \ x \in D_e. x \ \text{snores}.
\]

\[
F(\text{child}_{et}) = \lambda x. x \in D_e. x \ \text{is a child}.
\]

\[
F(\text{hates}_{et}) = \lambda x. x \in D_e. \ y \in D_e. y \ \text{hates} \ x.
\]

\[
F(\text{obnoxious}_{et}) = \lambda x. x \in D_e. x \ \text{is obnoxious}.
\]

\[
F(\text{every}_{et}) = \lambda P.P \in D_{et}. \lambda Q.Q \in D_{et}. (\forall x) (P(x) \rightarrow Q(x))
\]

\[
F(\text{some}_{et}) = \lambda P.P \in D_{et}. \lambda Q.Q \in D_{et}. (\exists x) (P(x) \land Q(x))
\]

\[
F(\text{no}_{et}) = \lambda P.P \in D_{et}. \lambda Q.Q \in D_{et}. (\neg \exists x) (P(x) \land Q(x))
\]

The semantic meta-language uses obvious abbreviations. In particular, I am using H\&K’s convention that meta-linguistic sentences stand for the truth-value \( 1 \) if they are true, for the value \( 0 \) otherwise. In what follows I will even use a shorter notation and write \( \lambda f_a \) for \( \lambda f.f' \in D_a \).
Given the semantic lexicon, we are in a position to state a recursive interpretation \( [[\cdot]] \) for the entire language. This function depends on the structure \( M = (E, \{0,1\}, F) \) and a variable assignment \( g \). \( M \) is called a model.

(17) The interpretation \( [[\cdot]]^{M,g} \) for EL

1. Let \( \alpha \) be a lexical tree of type \( a \). Then \( [[\alpha]]^{M,g} = F(\alpha) \). (L)

2. Let \( x \) be a variable of type \( a \). Then \( [[x]]^{M,g} = g(x) \). (Var)

3. Let \( \gamma \) be a branching tree of type \( b \) with daughters \( \alpha \) of type \( ab \) and \( \beta \) of type \( a \). Then \( [[\gamma]]^{M,g} = [[\alpha]]^{M,g} ([[\beta]]^{M,g}) \). (FA)

4. Let \( \gamma \) be a branching tree of type \( ab \) with daughters \( \alpha \) and \( \beta \) both of type \( ab \). Then \( [[\gamma]]^{M,g} = \lambda x \in D_a. [[\alpha]]^{M,g}(x) \& [[[\beta]]^{M,g}(x)] \). (PM)

5. Let \( \alpha \) be a tree of type \( ab \) of the form \( [ab x] \), \( x \) a variable of type \( a \) and \( \beta \) a tree of type \( b \). Then \( [[ab \lambda x\beta]]^{M,g} = \lambda u \in D_a. [[\beta]]^{M,g}[x/u] \) (\( \lambda \))

The most complicated interpretation principle is the abstraction rule \( \lambda \). We will illustrate it below. \( g[x/u] \) is a variable assignment that is defined like \( g \) with the possible exception that \( g[x/u](x) = u \). (There are other notations: H&K use \( g[x \rightarrow u] \), sometimes we find \( g[x\backslash u] \) and so on.) The relativisation of the interpretation function to a model is omitted if the model is self-understood from the context. Similarly, reference to the assignment function \( g \) is ignored if we don’t need it. The LF in (15) can now be interpreted along the lines indicated in section 2.

4.3. Interpretations of External Merge

Here is a first overview of the relation of languages of the EL-type and those of the MP-type. This comparison reveals which kind of interpretation rules (rules of construal) we have to expect.

The rule External Merge (EM) combines two expressions \( \alpha + \beta \) to a new expression \( \gamma \). The following cases may arise.

(a) One of the two expresses a function and the other expresses an argument of the right type. In this case the Interpretation is Functional Application. This is the case if Mary + snores combine to a sentence by EM.

(b) The two expressions are of the same type, say et. In this case EM is interpreted as Predicate Modification. An example will be the combination of an NP with a relative clause, e.g. woman + that snores.
(c) One of the two expressions is semantically empty. In this case, we delete the semantically void expression at LF. An example is the preposition of in the mother of Mary. mother expresses a function of type e(et). Mary has the type e and is a good argument. The preposition of doesn’t add anything to the content and is therefore deleted at LF.

(d) The two expressions have types that fall neither under (a) nor (b). In this case we need a special interpretation rule. We won’t be concerned with such cases in this article. Cases like this are the domain of generalized categorial grammar.

4.4. Interpretation of Internal Merge

The interpretation of Internal Merge (IM) will deal with the following cases.

1. A head is moved, e.g. a verb. In this case the meaning of the expression is not affected and the head is reconstructed to its base position at LF.

2. An argument expression is moved. If the argument is of the logical type required by the argument position, the moved expression might be reconstructed at LF.

3. The argument position is of type e and the expression located there is of type (et)t. In this case, the argument is moved for type reasons. This movement will create a trace of type e and an abstract of type et. The interpretation of this kind of movement will be the rule QR (Quantifier Raising). QR will typically be a covert movement and apply at the LF branch. QR will play a central role in this survey.

4. An argument position is occupied by a semantically empty pronoun. This is an expression without meaning and without logical type, often called PRO. This expression is moved at the left periphery and creates a λ-abstract. We will call this kind of movement PRO-movement. It can be used to create any kind of λ-abstract. PRO movement plays an important role for the treatment of quantifiers in DPs or PPs. It is also used for question formation and plays a central role for the interpretation of tense and aspect. At LF, PRO is deleted. PRO originates in a case-less position. Semantically empty pronouns that are base-generated at a case position are called WH(-operators). The movement of WH is called WH-movement. At typical instance of WH is the relative pronoun. Semantically, there is no difference between PRO and WH. Both are semantically void.

Notes on the literature: There are many versions of typed languages. The versions used in linguistics, especially the semantics for λ-abstraction, go perhaps back to different writings of Montague, e.g. (Montague, 1970c). The tradition of these languages is, however
much older and goes back to Frege’s abstraction of the graph of a function, Russell’s type theory and so on. Montague uses an intensional system (see below). An influential extensional system is the Ty2 language by (Gallin, 1975).

5. **THE TWO MOST IMPORTANT RULES OF CONSTRUAL: FI AND QR**

This section introduces two important interface principles: the *Principle of Full Interpretation* (FI) and the rule *Quantifier Raising* (QR). We will discuss three tasks performed by QR: (a) resolutions of type conflicts, (b) spelling out scope ambiguities of a quantifier, and (c) variable binding.

Let us see first how the LF for *Mary snores* is obtained from the syntax. Recall that lexical trees come from the syntax with both a syntactic category and a logical type. (More accurately, the lexicon contains for each word a pair of lexical trees \(<\alpha, \beta>\), where \(\alpha\) is for the syntax and \(\beta\) is for the LF.)

(18) \[ SS: [V [N,e Mary] [V,et snores]] \]

The syntactic categories \(V\) and \(N\) are not relevant for the LF-language. Therefore we delete them and obtain the intermediate sequence of lexical trees:

\[ [e Mary] [et snores] \]

The rule FA tells us that these combine to a tree of type \(t\), i.e., the derived LF is the tree

(19) \[ LF: [t [e Mary] [et snores]] \]

The personal ending -s is not important for the interpretation and may be deleted as well (but we ignore this possibility). We call the principle that is responsible for the deletion of semantically inert material the *Principle of Full Interpretation* (FI).\(^5\)

(20) **FI**: An LF tree contains only material that is important for the semantic interpretation. (Similarly a PF tree only contains material that is important for phonetic interpretation.)

**FI** is the first important interface principle.

5.1. **QR resolves type-clashes**

Let us construct next an LF for the SS in (9), here repeated as:

(21) \[ [S [NP Bill] [VP [V hates] [DP [Det every] [NP [AP obnoxious] [N child]]]]] \]

\(^5\) As far as I know the principle originates in Chomsky’s work, perhaps in (Chomsky, 1986).
We assume the following lexical rules for the generation with EM.

(22) Some lexical entries

\[
\begin{align*}
N_e & \rightarrow \text{Bill, Mary} \\
N_{et} & \rightarrow \text{boy, child, student} \\
V_{et} & \rightarrow \text{sleeps, snores} \\
V_{e(et)} & \rightarrow \text{likes, hates} \\
\text{Det}_{(et)(et)} & \rightarrow \text{every, a, no} \\
A_{et} & \rightarrow \text{nice, obnoxious}
\end{align*}
\]

If we proceed according to the principle FI, i.e., we delete the syntactic categories and try to project the logical types according to the rules of the LF-syntax, we get as far as this:

(23) A type clash

![Diagram of type clash]

There is no way to assign a type to the VP-node in our system. The daughters of that node can neither be combined by FA nor by PM. This is due to the fact that the object is a generalised quantifier, i.e., a function that assigns truth-values to sets of individuals. H&K call this the problem of the object. Before we solve this problem, we calculate the meaning of the object because the calculation illustrates the usefulness of the rule PM.

\[
\begin{align*}
&\text{[[}\text{et}\text{ every}_{(et)(et)}\text{ et obnoxious}_{et}\text{ child}_{et}\text{ ]}]\text{ ]} \\
&= \text{[every}_{(et)(et)}\text{ ]}([\text{et obnoxious}_{et}\text{ child}_{et}\text{ ]}) \quad \text{FA} \\
&= [\lambda P_{et}.\lambda Q_{et}.(\forall x)(P(x) \rightarrow Q(x))]\text{ }([\text{et obnoxious}_{et}\text{ child}_{et}\text{ ]}) \quad \text{L} \\
&= [\lambda Q_{et}.(\forall x)([\text{et obnoxious}_{et}\text{ child}_{et}\text{ ]})(x) \rightarrow Q(x))] \quad \lambda\text{-conversion} \\
&= [\lambda Q_{et}.(\forall x)(\lambda y[ \text{et obnoxious}_{et}\text{ ]}(y) & \text{et child}_{et}\text{ ]}(y))(x) \rightarrow Q(x))] \quad \text{PM} \\
&= [\lambda Q_{et}.(\forall x)(\lambda y[ \text{et obnoxious}_{et}\text{ ]}(y) & \text{et child}_{et}\text{ ]}(y))(x) \rightarrow Q(x))] \quad \lambda\text{-conversion} \\
&= [\lambda Q_{et}.(\forall x)(\lambda y[ (y\text{ y is obnoxious})(x) & [\lambda y.x \text{ y is a child}](x)) \rightarrow Q(x))] \quad \text{L: } 2\times \\
&= \lambda Q_{et}.(\forall x)(x\text{ is obnoxious} & \text{ x is a child} \rightarrow Q(x)) \quad \lambda\text{-conversion: } 2\times
\end{align*}
\]
This is the correct meaning, viz. the property true of a set if every obnoxious child is a member of that set. The derivation illustrates what the rule Predicate Modification (PM) achieves: it is set-theoretical intersection.

Since we know that there is a compositional interpretation for the object in (23), we can simplify the structure by letting the general quantifier unanalysed, i.e., we assume that we have a lexical \([\text{et}t\ \text{every obnoxious child}]]\), which has the meaning just calculated.

The type clash is resolved by May’s rule Quantifier Raising (QR); cf. (May, 1977, May, 1985), which creates a \(\lambda\)-abstract by movement.

(24) The rule QR
Move an NP or DP out of an XP and adjoin it to XP. Leave a co-indexed trace. If \(i\) is the index created by the rule, the trace \(t_i\) is interpreted as a variable of type \(e\), the movement index \(i\), i.e., the index of the moved NP/DP, is spelled out as a \(\lambda i\) at LF.

QR may be regarded as an instance of Internal Merge. We apply the rule before we delete uninterpretable stuff by FI. Here is a picture that illustrates the solution of the problem of the object by means of QR:

(25)

We see that the types project according to the syntactic rules of the LF-language. We delete the syntactic categories, interpret the movement index \(5\) as \(\lambda 5\) and the trace \(t_5\) as a variable of type \(e\). Thus we derive the LF-tree:

(26) \([t \ [\text{et}t\ \text{every obnoxious child}]] \ [\text{et} \ [\text{Bill}_e \ [\text{et} \ \text{hates}_{et(t}_{5}\ t_5)]]]]\]

To see how the abstraction rule (\(\lambda\)) works, let us calculate the meaning of the \(\lambda\)-abstract. Let \(g\) be any variable assignment.

\[\llbracket [\text{et} \ldots] \rrbracket^g\]

\[= \lambda u_e \llbracket [\text{et} \ [\text{Bill}_e \ [\text{et} \ \text{hates}_{et(t_{5}\ t_5)\ t_5}]]]]^u_{t_5} \]

\(\lambda\)-rule: Variable binding!
The essential step in the computation, i.e. the binding of the variable \( t_5 \) is achieved by the \( \lambda \)-rule. The rest of the computation is straightforward.

Now we combine the two immediate constituents of the LF in (26) by means of \( \text{FA} \). In other words, we have for an arbitrary assignment \( g: \)

\[
\begin{align*}
\llbracket [\llbracket (e(e) t_5) \text{ every obnoxious child} \rrbracket [\llbracket (e(e) [\llbracket (e(e) t_5) \text{ Bill} \rrbracket [\llbracket (e(e) t_5) \text { hates} \rrbracket ] ] ] ] \rrbracket \rrbracket & = \llbracket [\llbracket (e(e) t_5) \text{ every obnoxious child} \rrbracket \llbracket (\llbracket (e(e) (e(e) t_5) \text { Bill} \rrbracket [\llbracket (e(e) t_5) \text { hates} \rrbracket ] ] ] \rrbracket \rrbracket ] \text{ FA} \\
\end{align*}
\]

We substitute the two meanings computed before and obtain:

\[
= \llbracket (\forall x)(x \text{ is obnoxious & } x \text{ is a child } \rightarrow Q(x))(\lambda u_3[\text{Bill hates } u]) \text{ previous calculation} \\
= (\forall x)(x \text{ is obnoxious & } x \text{ is a child } \rightarrow \text{ Bill hates } x) \quad \lambda \text{-conversion 2x}
\]

Thus the rule \( QR \) solves the problem of the object. More generally, the rule \( QR \) helps to overcome many type clashes arising with DPs that expresses generalised quantifiers. \( QR \) is therefore a further very important interface principle mediating between SS and LF.

5.2. \( QR \) and Scope Ambiguities

The best-known job done by \( QR \) is the spelling out of quantifier scope. Consider Cresswell’s (1973) favourite sentence:

(27) Everyone loves someone.

It has two readings:

(28) a. \( (\forall x)[x \text{ is a person } \rightarrow (\exists y)[y \text{ is a person } \text{ & } x \text{ loves } y]] \)

b. \( (\exists y)[y \text{ is a person } \text{ & } (\forall x)[x \text{ is a person } \rightarrow x \text{ loves } y]] \)

The two quantifiers involved have the type (et)t. Therefore the SS exhibits the same type clash as before. We resolve the conflict by \( QR \). \( QR \)-ing the object gives us the following structure:

(29) \[ [\llbracket \text{DP someone}_{(e(t))} \rrbracket [\llbracket \text{DP everyone}_{(e(t))} \llbracket [\text{VP [loves}_{(e(t))} \text{ t}_1]]] ] \] (from SS by QR)
The deletion of the syntactic categories by FI and the projection of the types according to the LF-syntax gives us an LF with the reading in (28b). Note that we did not have to QR the subject because we can apply the subject to the VP.

If, however, we QR the subject in a second step and adjoin it to the entire structure, we obtain an LF that expresses the reading in (28a):

\[
[ \text{DP} \text{everyone} \text{et} t_2 \text{[DP someone} \text{et} t_1 \text{S} t_2 \text{[VP} \text{loves} \text{et} t_1 \text{t_1]]}}]
\]

(from SS by QR: 2x)

=> (FI) \[
[i[ \text{everyone} \text{et} t_2 i \lambda t_2 [[\text{someone} \text{et} t_1 i \lambda t_1 \text{et}[\text{loves} \text{et} t_1 \text{t_1]]]}]]
\]

The somewhat tedious computation of the meanings of the LFs is left to the reader. We presuppose obvious interpretations for the words everyone, someone and loves.

A note to the terminology scope: The scope of a phrase is its c-command domain. A quantifier \(\alpha\) has wide scope with respect to a quantifier \(\beta\) ("\(\alpha\) out-scopes \(\beta\)") if \(\alpha\) c-commands \(\beta\). At SS, everyone has wide scope with respect to someone. The LF in (30) preserves the scope relation. But the LF in (29) reverses the relative scope: someone out-scopes everyone.

5.3. \(QR\) binds pronouns

To illustrate pronoun binding by QR, consider the following sentences with their intended interpretations:

(31) \textbf{Every actor}_1 \textbf{admires himself}_1

\((\forall x)[\text{if x is an actor, x admires x}]\)

(32) \textbf{Every student}_1 \textbf{owns the computer in front of him}_1

\((\forall x)[\text{x is a student } \rightarrow \text{ x owns the unique y[y is a computer & y is in front of x]}]\)

The co-indexing represents binding. A notation like this is used in the GB-theory and its followers, but it is not explained there. Let us interpret it. The pronouns himself\(_1\) and him\(_1\) both represent the variable 1 of type e and hence have the same meaning, viz. they denote the individual \(g(1)\), where \(g\) is the relevant variable assignment. The difference in use come from different binding conditions, to which we will return in due time.

Here is the SS of first sentence:

(33) \[
[S \text{[DP} \text{et} \text{every actor}_1 i \quad \text{VP} \text{et} \text{admires} \text{et} \text{him}_1]]
\]

Up to the VP, there is no type conflict. If we ignored the index of the subject, we could
combine the subject with the VP by means of FA. That would yield the wrong reading, viz. 
\((\forall x)[x \text{ is an actor } \rightarrow x \text{ admires } g(1)]\). Suppose \(g(1) = \text{Bill}\), then the LF would be true if every actor admires Bill. The sentence cannot have this meaning.

Recall H&K’s interpretation of movement indices: they are regarded as \(\lambda\)-operators. Let us adopt the convention, that an index is interpreted as a \(\lambda\)-operator if it cannot be a “referential” index of a variable, i.e. a diacritic that distinguishes the variable from others of the same type. The indices of non-pronominals (e.g. quantifiers or names) are interpreted as \(\lambda\)-operators.

Returning to (33) it appears that we haven’t gained anything by this move. Suppose, the index 1 of every actor were \(\lambda 1\). Then the \(\lambda\)-abstract \([\varepsilon(et) \lambda 1 [VPet [Ve(et) \text{ admires} [NPe \text{ himself}_1]]]]\) would have the type of a transitive verb. This generates our familiar type clash. So this is not an option. Suppose therefore, we adopt the convention that the index of every actor is regarded as a movement index: it tells us that the DP has to be QR-ed and has that index after movement. In other words, the \(\lambda\)-operator is already there, but the DP has not yet been moved. According to this recipe the SS in (33) is transformed into the following structure by QR:

\[
(34) \quad [[\text{DP(et)} \text{ every actor}]_1 [S t_1 [VPet [Ve(et) \text{ admires} [NPe \text{ himself}_1]]]]]
\]

Applying FI and projecting the types, we obtain the desired LF:

\[
(35) \quad [[[\varepsilon(et) \text{ every actor}]_1]_1 \lambda[1_1 t_1 [\varepsilon(et) \text{ admires} [\varepsilon(\text{him}_1)]]]]]
\]

The reader may compute for himself that the LF is true if the condition in (31) holds. In particular, he should check that the \(\lambda\)-abstract expresses the property \([\lambda u_1. u \text{ admires } u]\). If we apply the meaning of the generalised quantifier expressed by the subject to this, the claim follows immediately.

The SS in (32) is construed along the same lines. The LF is something like this:

\[
(36) \quad [[\varepsilon(et) \text{ every student}]_1 \lambda[1_1 t_1 \text{ owns } [\varepsilon(\text{the}) \varepsilon(\text{computer}) [\varepsilon(\text{in_front of } \text{him}_1)]]]]]
\]

We QR the subject and thereby bind the variable \text{him}_1. The rule for the definite article is this:

\[
(37) \quad \text{The definite article}
\]

\(\text{the}\) has the type \((et)e\). \(F(\text{the})(P_{et})\) is only defined if \(P\) is a singleton. If defined, \(F(\text{the})(P) = \text{the unique element of } P\).

The modifier \([\varepsilon(\text{in_front of } \text{him}_1)\] is combined with the head noun \text{computer} by PM. I am
assuming the preposition of is semantically empty and therefore deleted at LF. The semantic lexicon entry of the complex preposition in_front is this:

\[(38) \quad [P, \text{of} \text{et} \text{in} \text{front}] \text{selects a PP with head of. of has no type and no meaning. (That’s why it doesn’t deserve a bold face print.)}\]

\[F([P, \text{of} \text{et} \text{in} \text{front}]) = \lambda x_e \lambda y_e.y \text{ is in front of } x.\]

The SS \([P, \text{of} \text{et} \text{in} \text{front}]\ [PP \text{of} \text{P} [NPe \text{him} 1]]\) is transformed into \([e \text{et} \text{in} \text{front}] [e \text{him} 1]\) by FI. The reader may convince himself that the LF in (36) expresses indeed the truth-condition stated in (32). Again, a full computation is tedious.

**QR** applies on the way to LF. At SS we have the DPs in argument positions, where they might cause a type conflict. QR solves the type conflict. After the application of QR we delete semantically vacuous material according to FI and project the types according to the rules of the logical syntax.

Summary: In this section we have shown that QR serves three purposes: (a) it solves (certain) type clashes; (b) it spells out scope ambiguities arising with more than one quantifier; (c) it performs pronoun/variable binding. In section 15 we will ask whether we really need the rule QR. The problem of the object can be solved by type-lifted lexical entries. Pronoun binding can be done by special rules of composition. But QR is still needed for spelling out certain scope ambiguities. The rule QR is the most transparent method for solving the three problems and all the alternative methods hide the rule QR somewhere else.

Notes to the literature: QR has been introduced into the literature in (May, 1977). May doesn’t give a semantics for the rule, however. The only use he makes of QR is the spell out of quantifier ambiguities. The present semantics has been developed be Irene Heim in lecture notes. Montague has a rule of Quantifying-in, which does the same job with a different syntax and precise semantics (cf. the Rule S14a in (Montague, 1970c)). The term Generalised Quantifier has become popular through (Barwise and Cooper, 1981). (Ajdukiewicz, 1935: last page) is presumably the inventor of the Generalised Quantifier.

6. **RELATIVE CLAUSE AND QUANTIFYING INTO XP: THE EMPTY PRONOUNS WH AND PRO**

In this section we introduce semantically *empty pronouns*, which have no meaning and no type. Therefore they must be moved for type reasons. The semantics of QR has the effect that these pronouns turn out to be \(\lambda\)-operators at LF. The empty pronouns studied here are the relative pronoun **WH** and the empty subject **PRO** of VPs, NPs, PPs, and APs.
There is a long tradition among semanticists to interpret moved relative pronouns as λ-operators that bind a variable in the base position of the pronoun. H&K have proposed a theory that can explain the construction. We assume that *relative pronouns*, abbreviated as WH, are semantically vacuous, i.e. they have no meaning and no type. When we move a relative pronoun, it leaves a co-indexed trace of type e. By FI we delete the pronoun at LF and are left with the adjoined index, the wanted λ-operator. Consider the SS of the relative clause in the following example:

(39) Bill is a boy [CP WH4 C Mary likes t4]

FI gives us the LF [λ4 [λ Mary (λ likes(t4) t4)]], which expresses the property of being liked by Mary. We combine it with boy by PM. The final LF will involve QR of the object. (is expresses identity). The rule that moves WH, i.e. who, to [Spec,CP] is sometimes called operator movement. Apart from the different landing site it has the same characteristics as QR. It is an instance of Internal Merge.

The same method of construal applies to the covert pronoun PRO. In generative grammar, PRO is used as the subject of an embedded infinitival as in:

(40) Bill wants [CP C [VP PRO to sleep]]

Here PRO functions as the subject of a VP. We will take up such constructions later. H&K launch the natural generalisation that PRO may be the subject of all the other lexical projections as well (NP, AP, PP). Like relative pronouns PRO is semantically vacuous. This theory easily explains the occurrence of quantifiers within such phrases.

(41) Every owner of a Macintosh hates Windows. (Quantifying into NP)

Note first that the N-object a Macintosh must have narrow scope with respect to the determiner every. If QR gave the quantifier wide scope with respect to every, we would obtain a reading that is at best marginal, viz.

(42) ? (∃x)[Mac(x) & (∀y)[y is an owner of x → y hates Windows]]

The reading we are after is:

(43) (∀y)[(∃x)[Mac(x) & y is an owner of x] → y hates Windows]

Here is the SS of the subject DP with PRO as the NP subject.

(44) [DP every [NP PRO [NP owner(e0) [PP of [DP(e0) a Mac]]]]](SS)

Ignoring the intervening preposition of, we encounter the familial type clash between the
head noun owner and the object a Mac.

The structure is obtained by QR-ing a Mac first. In the next step PRO is QR-ed and has wide scope with respect to a Mac. Both movements leave a trace of type e. The next step consists in applying FI: (a) we delete PRO, leaving its index as λ₁; (b) we delete the vacuous preposition of; we delete the syntactic categories and project the types according the syntax rule of EL. This gives us a fully interpretable generalised quantifier:

(45) \[\text{every } [\lambda_{\_1} [\text{a Mac } [\lambda_{\_2} [t_1 \text{ owner } t_2]]]]\]

I leave it to the reader to add the types. We interpret owner exactly as it were the transitive verb owns. The reader may convince himself that this structure expresses the generalised quantifier \(\lambda P_{\_1} (\forall y) (\exists x) [\text{Mac}(x) \& y \text{ is an owner of } x] \rightarrow P(y)]\).

An example requiring Quantifying into PP is analysed in H&K, p. 221 ff.:

(46) Every student from a foreign country is admitted.

We analyse it exactly as before, i.e., the input for the rules of construal is the following SS for the subject:

(47) every student [PP PRO from_{\_1} a foreign country]

An example involving an adjective might be the following:

(48) No guest [AP PRO dissatisfied_{\_1} [PP with something]] complained

= \(\neg (\exists x) [x \text{ is a guest } \& (\exists y) [y \text{ is a thing } \& x \text{ is dissatisfied with } y] \& x \text{ complained}]\)

Here with is a semantically empty preposition.
To summarise this section: the grammar has to assume semantically empty pronouns, which may be overt (the relative pronoun *WH*) or covert (*PRO*). These must be QR-ed for type reasons. Semantically empty elements thus provide a close link between the grammar of natural languages and that of \(\lambda\)-abstraction. The method helps to understand a large array of constructions.

Notes on the literature: Relative pronouns have always been interpreted as \(\lambda\)-operators by semanticists. The tradition goes back at least to (Quine, 1960: p. 121). It remained, however, quite mysterious how the \(\lambda\)-operator was connected with WH-movement. The puzzle was solved by H&K’s theory of semantically empty pronouns. In (Heim and Kratzer, 1998), the authors develop such a theory for *PRO* only, but the extension to relative pronouns is obvious.

7. **INTENSIONAL CONTEXTS: THE TYPE LANGUAGE IL**

7.1. **Intension and Extension**

The language EL correctly describes whether a declarative sentence is true or false in our world, but it cannot grasp sentence meanings in general. The reason is very simple: there are only two truth-values but infinitely many sentence meanings. As said in the beginning, sentence meanings must be more complicated entities, viz. propositions. Most semanticists are adherents of possible world semantics according to which a sentence expresses a set of possible worlds (or the characteristic function of such a set). Consider the following sentences:

(49) a. *Bill believes that Mary is obnoxious.*

b. *Bill doesn’t believe that Mary snores.*

Suppose both sentences are true. Suppose further that both complement clauses are true. In the language EL the two complement clauses would then have the same meaning and Bill would be an irrational person because he would believe and not believe the same thing at the same time. But the scenario is perfectly possible without Bill being irrational. We have indicated Frege’s solution for this problem: in certain embedded contexts a sentence doesn’t express a truth-value ("Bedeutung") but a proposition ("Sinn"). Here are the propositions expressed by the two complements:

(50) a. \([\text{Mary is obnoxious}] = \lambda w. \text{Mary is obnoxious in } w.\)

b. \([\text{Mary snores}] = \lambda w. \text{Mary snores in } w.\)
A proposition reconstructs what Wittgenstein called the *truth-condition* of a sentence: We know the meaning of a sentence when we know under which conditions it is true. Think of possible worlds as scenes presented on a TV screen. We want to find out of some person whether he knows the meaning of *Mary snores*. We show him scenes in which Mary snores and others in which she doesn’t. The test person has to answer the question: Is the sentence *Mary snores* true in this scene or not? If he gives the correct answer for each scene, we conclude that he has understood the sentence. In fact, the function in (50b) assigns 1 to every scene in which Mary snores and 0 to every scene in which she doesn’t. So this notion of proposition is a reasonable approximation to the truth-condition of a sentence.

Sentence meanings like those in (50) are called *intensions*. Not only sentences have intensions, but every expression of the language has one be it basic or complex. The language IL is designed to reconstruct that idea. The name IL reminds of Montague’s *Intensional Logic*; cf. (Montague, 1970b, Montague, 1970c, Montague, 1973). The system used here has, however, a slightly different syntax from Montague’s IL. It is the system sketched in chapter 12 of H&K.

### 7.2. Syntax and Semantics of IL

The initially rather confusing feature of the language IL is that the types are largely as before but they encode something different, namely *intensions*. For instance the meaning of *snores* will be the intension \([\lambda w \in W. \lambda x. x \text{ snores in } w]\), where W is the set of possible worlds. Only when we apply the verb meaning to a particular world w, we obtain an *extension*, viz. the set \([\lambda x. x \text{ snores in } w]\). I will say something more about this in a moment.

Let us introduce the syntax first.

The type system for the language is slightly richer. There are the types of EL plus a new type s (“sense”) that can be prefixed to any EL-type producing an IL-type. The type grammar is therefore this:

(51) The type system for IL

The basic types are e and t. The complex types are generated by two rules:

(a) If a and b are types, then (ab) is a type.
(b) If a is a type, (sa) is a type.

The system of semantic domains belonging to IL is the following one:

(52) a. \(D_e = E = \) the set of individuals
b. \(D_{(ab)} = D_b^{Da}\)
c. $D_{(sa)} = D_a^W$, where $W$ is the set of possible worlds.

The functions in $D_{sa}$ are called *a-intensions*. If $f$ is an a-intension and $w$ is a possible word, $f(w)$ is the *extension of $f$ in $w$*.

The syntax of IL is more or less the same as that of EL. The only innovation is the rule IFA, which is due to H&K.

(53) *The syntax of IL*

consists of the same rules as the syntax of EL with the addition of the following rule:

If $\alpha$ is an IL-tree of type $(sa)b$ and $\beta$ is an IL-tree of type $a$, then $[b \alpha\beta]$ is an IL-tree of type $b$. \hspace{1cm} (IFA)

The rule is called *Intensional Functional Application*.

The interpretation of IL depends on a semantic lexicon, i.e. a function $F$ that interprets the lexical trees.

At this step we have to be aware of the *Main Characteristics of IL: the meanings of expressions of type $a$ are a-intensions*. In other words, if $\alpha$ is of type $a$, then $[\alpha]$ will not be in $D_a$ but in $D_{sa}$! Here are some entries of our EL-lexicon in (16) adapted to this IL-requirement. (Instead of “$\lambda w \in W$” I will write “$\lambda w$”.)

(54) Some entries of the semantic lexicon

$F(\text{Mary}) = \lambda w.\text{Mary}$

$F(\text{snores}_{et}) = \lambda w.\lambda x_e. x \text{ snores in } w.$

$F(\text{child}_{et}) = \lambda w.\lambda x_e. x \text{ is a child in } w.$

$F(\text{hates}_{et}) = \lambda w.\lambda x_e.\lambda y_e. y \text{ hates } x \text{ in } w.$

$F(\text{obnoxious}_{et}) = \lambda w.\lambda x_e. x \text{ is obnoxious in } w.$

$F(\text{every}_{et}(et)((et)t)) = \lambda w.\lambda P_{et}.\lambda Q_{et}. (\forall x)(P(x) \rightarrow Q(x))$

In order to be able to interpret the sentences in (49), we add the entries for *not* and *believes*.

(55) $F(\text{not}_{et}) = \lambda w.\lambda x_e.1, \text{ if } x = 0; 0, \text{ if } x = 1.$

$F(\text{believes}_{et}(et)(et)) = \lambda w.\lambda p_{et}.\lambda x_e. x \text{ believes } p \text{ in } w.$

The interpretation for *believes* looks rather trivial because it does not analyse what it means to believe something. Following (Hintikka, 1969), the standard analysis for *believe* is that of a verbal quantifier: “$x \text{ believes } p \text{ in world } w$” is true if $p$ is true in every world $w'$ compatible with what $x$ believes in $w$. For our purposes this analysis is not relevant. The important point is that the object of *believe* has to be a proposition and not a truth-value. If
we look at the types of our lexical entries we see that the entry of *believes* is the only one that has an intensional type as an argument. Functors with such a type are called *intensional*. Functors that only have extensional types as arguments are *extensional*. The intensions expressed by extensional functors are constant functions.

Given the semantic lexicon, we are in a position to state a recursive interpretation $[[ . ]]$ for the language IL. This function depends on the model $M = (E, \{0,1\}, W, F)$ and a variable assignment $g$.

(56) **The interpretation $[[ . ]]^{M,g}$ for IL**

1. Let $\alpha$ be a lexical tree of type $a$. Then $[[ \alpha ]]^{M,g} = F(\alpha)$. \hspace{1cm} (L)
2. Let $x$ be a variable of type $a$. Then $[[ x ]]^{M,g} = \lambda w.g(x)$. \hspace{1cm} (Var)
3. Let $\gamma$ be a branching tree of type $b$ with daughters $\alpha$ of type $ab$ and $\beta$ of type $a$.

   Then $[[ \gamma ]]^{M,g} = \lambda w.([[[ \alpha ]]^{M,g}(w))([[ \beta ]]^{M,g}(w))]$. \hspace{1cm} (FA)
4. Let $\gamma$ be a branching tree of type $b$ with daughters $\alpha$ of type $(sa)b$ and $\beta$ of type $a$.

   Then $[[ \gamma ]]^{M,g} = \lambda w.([[[ \alpha ]]^{M,g}(w)([[ \beta ]]^{M,g})]$. \hspace{1cm} (IFA)
5. Let $\gamma$ be a branching tree of type $at$ with daughters $\alpha$ and $\beta$ both of type $at$. Then $[[ \gamma ]]^{M,g} = \lambda w.\lambda x.([[[ \alpha ]]^{M,g}(w)(x) \& [\beta ]]^{M,g}(w)(x)]$. \hspace{1cm} (PM)
6. Let $\alpha$ be a tree of type $ab$ of the form $[ab x]$, $x$ a variable of type $a$ and $\beta$ a tree of type $b$. Then $[[[ab x\beta]]]^{M,g} = \lambda w.\lambda u \in D_a.[[\beta ]]^{M,g[x/u]}(w)$ \hspace{1cm} (\lambda)

We are now in a position to analyse the sentences in (49). The SS of (49a) is this:

(57) \[
[S \text{ Bill}_e \text{ believes}_{(st)(et)} [CP \text{ that } [S \text{ Mary}_e [VP \text{ is } [AP \text{ obnoxious}_{et}]]]]]]
\]

I am assuming that “that” and “is” are semantically empty and hence deleted by FI. After type projection we obtain the following IL-tree:

(58) \[
[t \text{ Bill}_e [et \text{ believes}_{(st)(et)} [t \text{ Mary}_e [et \text{ obnoxious}_{et}]]]]
\]

At first sight it looks as if we had a type clash between the types of daughters of the matrix VP, because its daughters are of type $(st)(et)$ and $t$ respectively. These cannot be put together by the rule FA, but the rule IFA takes care of them. Let us see how this works. I leave it to the reader to convince herself that $[[t \text{ Mary}_e [et \text{ obnoxious}_{et}]]$ expresses indeed the proposition in (50a), i.e. we assume:

$[[ t \text{ Mary}_e [et \text{ obnoxious}_{et}]]] = \lambda w'.\text{Mary is obnoxious in } w'$. 

We calculate the meaning of the matrix VP:
Next we calculate the meaning of the entire sentence:

\[
\begin{align*}
&\lambda w[\lambda x.\lambda p.x \text{ believes } p \text{ in } w)(\lambda w'.\text{Mary is obnoxious in } w')]
&\quad\text{(FA)}
\end{align*}
\]

\[
\begin{align*}
&\lambda w[\lambda x.\lambda p.x \text{ believes } p \text{ in } w)(\lambda w'.\text{Bill believes } w').\text{Mary is obnoxious in } w']
&\quad\text{in } w]
&\quad\lambda\text{-conversion: } 2x
\end{align*}
\]

Let us say a word about the interpretation of the sentence in (49b). Here we face difficult problems of the English syntax of negation and do-support, which I don’t want to take up here. I am assuming that the relevant LF is the following IL-tree:

\[
\text{(59) } [\text{nottt } [\text{Bill believes } \text{Mary is obnoxious}]]
\]

The reader may calculate for himself that the interpretation gives the following proposition:

\[
\lambda w[1, \text{ if } [\text{Bill believes } \text{Mary is obnoxious} \text{ in } w'] \text{ in } w] = 0;
0, \text{ if } [\text{Bill believes } \text{Mary is obnoxious} \text{ in } w'] \text{ in } w] = 1
\]

This is the set of worlds in which Bill doesn’t believe that Mary is obnoxious. Note that this doesn’t capture what is called \textit{Neg-Raising} in the literature, i.e. the fact that the most natural interpretation of (39b) is the proposition that Bill believes that Mary is not obnoxious.

Another problem of the English syntax of negation is that a quantifier in subject position may be \textit{reconstructed} under the scope of negation though it has wide scope with respect to it at SS. The most natural reading of (60a) is paraphrased by (60b).

\[
\text{(60) } \begin{align*}
a. & \text{ Everyone doesn’t like Mary} \\
b. & \text{ Not everyone likes Mary}
\end{align*}
\]

So one possible LF for (60a) must be the structure [\textbf{not} [\textbf{everyone} [\text{likes Mary}]]]. Here, \textbf{everyone} is interpreted at its base position. The interesting question for reconstruction (or \textit{Quantifier Lowering}) is of course when it is allowed and when not. We will discuss more cases of reconstruction below.

It is revealing to compare the rule \textbf{IFA} with Richard Montague’s method to deal with
intensional operators. Montague’s IL contains a logical operator ^, which may be called up-operator. It converts an expression of type a into one of type sa. Here is the syntax and semantics.

(61) Montague’s ^-operator (“up-operator”)

If [\(a\)] is a tree of type a, then \([sa^a]\) is a tree of type sa.

\[\llbracket [sa^a] \rrbracket_M^{\delta} = \lambda w[\lambda w'[\llbracket [a] \rrbracket_M^{\delta}(w')] = \lambda w[\llbracket [a] \rrbracket_M^{\delta}] \]

Thus the intension of \(^a\) is a function that assigns any world w the intension of a. In other words, the extension of \(^a\) in world w is the intension of a. Montague assumes that that-clauses are translated into IL-trees prefixed with ^. If we do that, we don’t need the rule IFA, FA suffices. Let us see how this works. Let us assume that the LF of (49a) is not (58) but the following tree instead:

(62) \[[t \text{ Bill} \langle \text{et believes}\rangle \langle\text{et}\rangle [\text{st} \langle\text{et}\rangle [\text{st} ^{t} \langle\text{et}\rangle \langle\text{et}\rangle \text{Mary} \langle\text{et}\rangle \text{obnoxious}\langle\text{et}\rangle]]]]\]

The matrix VP doesn’t exhibit a type conflict, and we can use FA for the evaluation. The reader may compute for herself that the complement now expresses the following intension:

(63) \[\llbracket [\text{st} ^{t} \langle\text{et}\rangle \langle\text{et}\rangle \text{Mary} \langle\text{et}\rangle \text{obnoxious}\langle\text{et}\rangle]\rrbracket = \lambda w[\lambda w'.\text{Mary is obnoxious in } w']\]

Note that this is not the proposition that Mary is obnoxious. It is the intension that assigns any world the proposition that Mary is obnoxious. This is the correct meaning for computation of the VP-meaning via FA. The reader may check that the following evaluation is true:

\[\langle\text{et believes}\rangle \langle\text{et}\rangle [\text{st} ^{t} \langle\text{et}\rangle \langle\text{et}\rangle \text{Mary} \langle\text{et}\rangle \text{obnoxious}\langle\text{et}\rangle]\]

\[= \lambda w [\llbracket \text{believes} \rrbracket (w)(\llbracket [\text{st} ^{t} \langle\text{et}\rangle \langle\text{et}\rangle \text{Mary} \langle\text{et}\rangle \text{obnoxious}\langle\text{et}\rangle]\rrbracket (w))] \text{ FA}
\]

\[= \lambda w[\lambda x_\text{e} x \text{ believes } [\lambda w'.\text{Mary is obnoxious in } w'] \text{ in } w]\]

This is the same intension we had computed using the H&K strategy, i.e. IFA and no -operator. The answer is that both methods are equivalent. We may regard the ^-operator as a logical operation that helps to eliminate the type conflict between an operator of type (sa)\(b\) and an argument of type a. We prefix the argument with ^ and the conflict is solved. H&K’s IFA put the information that the functor does not apply to the extension of the argument but to the intension into the composition rule. The lesson to be drawn from this is that type conflicts do not exist as such. Whether the types of the daughters of a branching node are in conflict or not is a matter of the composition rules that determine the interpretation of the language. Montague’s system doesn’t have IFA, but it has the ^-operator instead. In what
follows, we will stick to H&K’s rule IFA.

Actually, Montague’s LF are much more complicated than H&K’s. The reason is that he generalises Frege’s dictum that an expression in an opaque context expresses a sense to every argument position. In his system, the verb *snores* has a type at least as complicated as (se)t. An article like *every* would have the type (s((se)t))(s((se)t))t. Generalised quantifiers have the type ((s((se)t))t). Montague takes it that verbs take generalised quantifiers as arguments. In (Montague, 1970b) he therefore proposes that intransitive verbs like *snore* have the type (s((s((se)t))t))t. It takes quite a bite of experience to indicate the semantic interpretations for verbs of that type and, accordingly, the formulas are very difficult to understand. There are other complications, e.g. the ‘-operator (“down operator”), which undoes the effect of the ^-operator. Features like these make Montague’s particular implementations of the LFs of English very hard to read. Nevertheless, the system has been popular a long time among semanticists, because Montague’s publications are the first serious theory of a semantic interpretation of natural language. The implementations proposed here are much simpler, but the semantic language used is basically Montague’s IL.

Here is the summary of this section. The meaning of complement clauses cannot be truth-values. They must be something more fine-grained, viz. propositions. This motivates the notion of intension as functions from worlds to extensions. The expressions of intensional languages denote intensions. We obtain an extension if the expression is evaluated at some world. We have two types of functional application. Extensional functional application (FA) is the application of a functor extension to an argument extension. Intensional Functional Application (IFA) is the application of the extension of a functor to the intension of the argument. If we introduce Montague’s up-operator into the syntax of IL, we only need FA as a principle of composition.

7.3. Notes on the literature
The idea that the meaning of a sentence can be identified with its truth-condition is due to (Wittgenstein, 1922/1984: 4.024). (Cresswell, 1991) takes difference in truth-conditions as the most certain principle for the non-identity of meaning. The historical roots for the development of intensional languages are (Carnap, 1947) and (Kripke, 1959). The version used here is due to chapter 12 of (Heim and Kratzer, 1998). H&K build on Montague’s publications, of course. The intension/extension dichotomy is due to the Frege tradition in the philosophy of language. Important systems that do it with a unified system of meanings (basically intensions without extensions) are (Lewis, 1972) and (Cresswell, 1973). The
detailed development of intensional logical languages and their application to the semantic interpretation is due to the work of Richard Montague; the milestones for the syntax/semantics interface are (Montague, 1970a), (Montague, 1970b), (Montague, 1973). Excellent pioneer work is (Cresswell, 1973). All of these works have come well over the years.

8. **Modality and Reconstruction**

In this section we motivate the need of *Reconstruction*, i.e., a quantifier or a head created by movement is interpreted at the position of its trace. Modals are raising verbs: the subject of a VP embedded under a modal is raised at SS to the subject position of the modal. If the subject is a generalised quantifier and it is interpreted at its SS-position, we obtain the so-called *de re* interpretation. But modal constructions have a second reading, the so-called *de dicto* interpretation. We obtain it, if we interpret the quantifier in the position of its trace. Thus we need Reconstruction as an additional rule of construal. Reconstruction is further needed for the interpretation of verb movement. An inspection of German particle verbs shows that verbs cannot be interpreted at their surface position in the general case.

8.1. **de re/de dicto**

Modals are quantifiers over possible worlds that are restricted by a contextually given set of worlds called *conversational background* (Kratzer) or *accessibility relation* (Kripke). They are raising verbs. If the subject of the modified VP (“the prejacent”) is a quantifier, we observe the so-called *de re/de dicto* ambiguity. The latter reading requires reconstruction of the quantifier on the way from SS to LF. *Reconstruction* therefore is a rule of construal.

Here is a modalised sentence with its SS and LF:

(64) You may/must be infected.
The meanings of the two modals are these:

(65) a. Possibility: $F(\text{may}) = \lambda w. \lambda R_s(\text{st}), \lambda p_s. (\exists w') [R(w)(w') \land p(w')]$

b. Necessity: $F(\text{must}) = \lambda w. \lambda R_s(\text{st}), \lambda p_s. (\forall w') [R(w)(w') \rightarrow p(w')]$

I am assuming that the subject originates as the subject of the adjective (or participle) infected. It is raised to [Spec, IP] by IM (“A-movement”). The interpretation of this movement is the same as that of QR. FI makes these structures fully interpretable. (We have to use IFA when we apply the modal to the prejacent.) With respect to a variable assignment $g$, the two LFs mean:

(66) a. $\lambda w. (\exists w') [g(R)(w)(w') \land \text{you are infected in } w']$

b. $\lambda w. (\forall w') [g(R)(w)(w') \land (\exists x) [x \text{ is a person in } w' \land x \text{ infected in } w']]$

Think of $g(R)$ as the function that assigns each world $w$ the symptoms you have in $w$ plus the proposition that you are infected if you have symptom $x$ and you are infected or drunk if you have symptom $y$. Consider a world $w_1$ in which you have symptom $x$. In this world both propositions are true. Next consider a world $w_2$ in which you have symptom $y$. In this world the proposition in (66a) is true but that in (66b) is false.

Next consider a sentence with a quantifier in subject position. It has the two readings indicated in (67a) and (b).

(67) Everyone may be infected.

a. $\lambda w. (\forall x) [x \text{ is a person in } w \rightarrow (\exists w') [g(R)(w)(w') \land x \text{ infected in } w']]$

b. $\lambda w. (\exists w') [g(R)(w)(w') \land (\forall x) [x \text{ is a person in } w' \rightarrow x \text{ infected in } w']]$

The first reading is called de re: the quantifier ranges over persons in the evaluation world,
the “actual” world. The second reading is called *de dicto*: the quantifier speaks about persons in the accessible worlds.

We obtain the *de re* reading immediately from the surface by **FI**:

\[(68) \quad \text{[everyone} \quad [\lambda_1 \quad [\text{may } R \text{ (st)}] \quad [t_1 \text{ infected}]]]\]

To obtain the *de dicto* reading, two methods are available for this example: (a) We could abstract over a variable of type (et)t. Then the quantifier would be *semantically reconstructed* by λ-conversion. (b) We move the quantifier to the position of its trace (*syntactic reconstruction* (**Rec**)). Here is the derivation of a reconstructed LF:

\[(69) \quad \text{a. [everyone} \quad [\lambda_1 \quad [\text{may } R \text{ (st)}] \quad [t_1 \text{ infected}]]]\]
\[\Rightarrow \text{ (Rec) } \quad [\text{may } R \text{ (st)}] \quad [\text{everyone infected}]]\]

In Chomsky’s MP, **Rec** is a rather natural process. Recall that the SS obtained by Internal Merge has not exactly the form in (69a) but rather the following one:

\[(70) \quad \text{[everyone}_1 \quad [\text{may } R \text{ (st)}] \quad \text{be [everyone}_1 \text{ infected}]]]\]

**FI** says that we have to delete uninterpretable stuff. One way of doing that is to delete the copy in base position leaving the index, which is interpreted as a variable. This gives us (69a). The other possibility is to delete the moved DP with its index. The index at the copy in base position doesn’t make sense semantically anymore, and it is therefore deleted. This gives us the *de dicto* LF in (69b).

### 8.2. Specific de dicto

There is a reading in modal constructions that cannot be obtained by means of the methods outlined. Consider the following sentence:

\[(71) \quad \text{A friend of mine must win.}\]

The scenario is this: My three friends Anna, Bertha and Cecile participate at the competition and I want one them to win no matter which one, I like them with equal love. If I give the DP a *friend of mine* wide scope with respect to the modal, I have wish about a particular friend. This is not the case. If I give it narrow scope, I seem to refer to the friends in the boulethic accessible worlds. There my friends might be very different from those in the actual world. The reading in question has been discovered by (Fodor, 1970)\(^6\) and has been called *specific de dicto*. (von Fintel and Heim, 2000: 6.1) point out that the reading is best

---

\(^6\) Fodor’s example is “Mary wanted to buy a hat just like mine”.

35
analysed by assuming a world pronoun \textbf{w-pro} as an argument of \textbf{a friend of mine}. \textbf{w-pro} has to be bound by stipulation. In other words, we must be able to refer to the hitherto implicit world argument in the syntax, i.e., we replace our intensional language by a purely extensional language. By convention we assume that the world argument is the first argument of a predicate. In order to form a proposition, von Fintel & Heim assume that these are formed by PRO-movement. The authors assume a W-PRO is generated at an i-position. W-PRO (but not \textbf{w-pro}!) is semantically empty and must be moved for type reasons generating a \(\lambda\)-operator. Here are some analyses:

(72) DS: \([t \text{ a friend of mine w-pro}_1 \text{ et wins W-PRO}]\)

\[\Rightarrow \text{PRO-movement (QR)}\]

LF: \(\text{W-PRO } \lambda_1[t \text{ a friend of mine w-pro}_1 \text{ et win t}_1]\)
\[= \lambda w.(\exists x)[x\text{ is a friend of mine in w & x wins in w}]\]

The representation of the \textit{de dicto} reading is this:

(73) \(\text{W-PRO } \lambda_2\text{ must(t}_2)(R)\text{ W-PRO } \lambda_1[t \text{ a friend of mine w-pro}_1 \text{ et win t}_1]\)

Here, \textbf{w-pro}_1 is bound by \(\lambda_1\) (and therefore indirectly by the modal). The \textit{de re} reading is generated by giving the indefinite term wide scope with respect to the modal:

(74) \(\text{W-PRO } \lambda_2\text{ a friend of mine w-pro}_2\lambda_3\text{ must(t}_2)(R)\text{ W-PRO } \lambda_1[t_3 \text{ et win t}_1]\)

This time the \textbf{w-pro} of the indefinite is bound by the matrix W-PRO and hence will refer to the actual world, if the proposition is applied to it.

We now see what the method buys: we can represent Fodor’s \textit{specific de dicto} reading by leaving the indefinite in the scope of the modal but binding its \textbf{w-pro} by the matrix W-PRO:

(75) \(\text{The specific de dicto reading}\)

\(\text{W-PRO } \lambda_2\text{ must(t}_2)(R)\text{ W-PRO } \lambda_1[t \text{ a friend of mine w-pro}_2 \text{ et win t}_1]\)

Note that our lexical entries haven’t changed, but the types have changed. The first argument of a predicate will always be a world. For instance, the rule for \textbf{wins} will be

(76) \(\text{F(wins}_{\text{det}}) = \lambda w.\lambda x.x\text{ wins in w.}\)

Fodor’s observation shows that something like this is the correct approach and the switch to an extensional language with world arguments is ultimately mandatory. We will however stick to our intensional language \textbf{IL} because most semanticists assume something like this.
8.3. Head movement

German verb movement provides evidence that syntactic reconstruction is more general than semantic reconstruction and hence to be preferred. German declaratives are formed by moving the finite verb to C. German has verbs with a prefixed particle, which is left back ("is stranded") when the finite part of the verb is moved. As an example, take the verb aufwacht "wakes up". It has the structure \([_{\text{V}} \ p \ \text{auf}] \ [_{\text{V}} \ \text{wacht}]\). Consider the following sentence with its standard analysis:

\[(\text{77}) \text{ German V2-movement} \]

\[\text{Ede wacht auf} \quad \text{"Ede wakes up"} \]

\[\begin{align*}
\text{CP}[\text{NP} \ \text{Ede}]_1 & \ [C'[[_{\text{V}} \ \text{wacht}]_2 \ C] \ [S \ t_1 \ [[p \ \text{auf}] \ t_2]]] \\
\end{align*}\]

It is not possible to interpret the trace semantically by assigning it the type of an intransitive verb \(\text{et}\). The reason is that the moved part of the verb has no meaning in isolation, neither has the particle \(\text{auf}\). The following semantic lexical entry is irreducible:

\[(\text{78}) \text{ A particle verb} \]

\[F([\text{et} \ [p \ \text{auf}] \ [_{\text{V}} \ \text{wacht}]]) = \lambda w. \lambda x. x \text{ wakes up in } w.\]

(I leave it open how exactly \(\text{FI}\) applies to the internal components of the particle verb. The two parts have no independent meanings, but we must be able to identify the verb. The problem doesn’t arise if we separate semantic entries from the syntactic ones.) It follows that we have to undo V-movement by a syntactic operation. Hence Syntactic Reconstruction is the more general method and I will assume that \((\text{Rec})\) is a rule of construal. Here is the construal of an LF for \((\text{77})\).

\[\begin{align*}
\text{CP}[\text{NP} \ \text{Ede}]_1 & \ [C'[[_{\text{V}} \ \text{wacht}]_2 \ C] \ [S \ t_1 \ [[p \ \text{auf}] \ t_2]]] & \text{SS} \\
\text{CP}[\text{NP} \ \text{Ede}]_1 & \ [C'[[_{\text{V}} \ \text{wacht}]_2 \ C] \ [S \ t_1 \ [[p \ \text{auf}] \ t_2]]] & \text{Rec} \\
[\text{Ede} & \ [\lambda t_1 \ [p \ \text{auf}] \ [_{\text{V}} \ \text{wacht}]])]] & \text{FI} \\
\end{align*}\]

We could have reconstructed the subject to the position of its trace. That would have produced the LF \([\text{Ede} \ [p \ \text{auf}] \ [_{\text{V}} \ \text{wacht}]])\), which has the same meaning.

Head movement requires reconstruction at least for German particle verbs. The common assumption is that head movement is always reconstructed.

Here is a summary of the results of the previous sections: Quantified subjects occurring in modal constructions are raised at SS. The interpretation by means of QR at this position provides the \textit{de re} reading. To get the \textit{de dicto} reading, we have to push the quantifier
back to the position of its trace. This is the rule of Reconstruction.

Fodor’s specific de dicto readings provide evidence that an intensional language of the IL-type is not sufficient for the analysis of natural language semantics. We need a language with world arguments in the syntax. At DS, the world positions are filled either with the pronoun \textit{w-pro} that has to be bound by stipulation or they are filled with the semantically empty \textit{W-PRO} that has to be moved for type reasons. \textit{W-PRO} is the semantic binder of \textit{w-pro}.

8.4. \textit{Notes on the literature:}

The semantics of modality was invented by (Kripke, 1959). The unification of Kripke’s semantics for an application to natural language is due to different publications by A. Kratzer, e.g. (Kratzer, 1978, Kratzer, 1977, Kratzer, 1981). The need for reconstruction is presumably first addressed in (May, 1985). The examples of this section are taken from (von Fintel and Heim, 2000). The theory of \textit{w-pro/W-PRO} is due to these authors. Evidence that something like this is necessary is much older; see e.g. (Heim, 1991). The typed language needed for the implementation of the theory is Gallin’s ty2-language; cf. (Gallin, 1975). The analysis of German V-movement is folklore and originates presumably in (Bierwisch, 1963). The classical reference for the syntax of head movement/incorporation is (Baker, 1988).

9. \textit{Questions: WH-Movement, Scope Marking, Pied-Piping}

9.1. \textit{Constituent Questions}

Questions involve both overt and covert WH-movement and are therefore interesting constructions for understanding the interface rules. In Karttunen’s (1977) theory of questions, the sentence in (79a) expresses the intension in (79b):

\begin{equation}
\text{(79)} \begin{align*}
\text{a. Which boy snores?} \\
\text{b. } \lambda w.\lambda p. \lambda x [\text{x is a boy in } w \& p = [\lambda w’.x \text{ snores in } w’] \& p(w)]
\end{align*}
\end{equation}

Suppose that Bill and John are the boys in \textit{w} that snore in \textit{w}. Then the question assigns \textit{w} the set of propositions \{that Bill snores, that John snores\}. The SS of the question is this:

\begin{equation}
\text{(80)} \begin{align*}
[\text{CP} \{\text{which boy}\}] \{c[\text{c } ?_{\text{adj}}(\text{adj}) \text{ WH-PRO}] [s t_{1} \text{ snores}]\}
\end{align*}
\end{equation}

The determiner \textit{which} has the same meaning as the indefinite article, in other words \textit{which boy} means the same as \textit{a boy}. The movement rule that places the wh-phrase to the position [Spec, CP] is called WH-movement. It has the same semantic characteristics as QR. The Minimalist Program claims that the movement is triggered by the need of checking an uninterpretable wh-feature in an adjacent position to the interrogative operator ?, which is
said to have an interpretable wh-feature. The meaning of ? is this:

\[(81) \text{ The question operator } ? \]

\[
F(?_{(st)(st)t}) = \lambda w. \lambda p_{st}. \lambda q_{st}. p = q
\]

The pronoun WH-PRO is semantically empty. (The notation might be confusing. There is only one PRO, but it might be generated at positions of a different type: so far we had the following one: PRO generated at an e-position without case is an empty subject; WH is the relative pronoun; it is generated at a case position of type e; W-PRO is generated at an s-position and generates a proposition; WH-PRO is generated at an st-position and generates sets of propositions. All these are nothing but λ-operators of a particular type. We could have named them PRO simpliciter.) In order to obtain an LF, we apply operator movement to WH-PRO and delete the uninterpretable material by FI. The result is this:

\[(82) \text{ A constituent question} \]

\[
[(st) WH-PRO_{2, st} [t \text{[which boy]} [et \lambda x [(st) ?_{(st)(st)t} t_{2, st} [t t_{1,e} \text{ snores}]]]]]
\]

The interpretation of the C’-node requires IFA. The reader may check that the LF gives us precisely the intension in (79b).

Apart from WH-PRO movement, the LF is almost the same as the SS. Multiple questions, however, require covert WH-movement, i.e. the rule belongs to the rules of construal as well. Consider the following question:

\[(83) \text{ Which girl likes which boy?} \]

The SS is as before with the object left in base position. The LF is generated by WH-moving the object to [Spec,CP]:

\[(84) \text{ A multiple question} \]

\[
[(st) WH-PRO_{2, st} [t \text{[which boy]} [et \lambda x [\text{[which girl]} [et \lambda y [(st) ?_{(st)(st)t} t_{2, st} [t t_{1,e} \text{ likes } t_{3,e}]]]]]]\]

\[ = \lambda w. \lambda p_{st}. (\exists x) [x \text{ is a boy in } w \text{ & } (\exists y) [y \text{ is a girl in } w \text{ & } p = [\lambda w’. y \text{ likes } x \text{ in } w’] \& p(w)]]
\]

9.2. whether-Questions

While the analysis of constituent questions is rather transparent, the analysis of whether-questions is more complicated and as far as I know there is no agreement in the literature on what the best analysis is. The analysis presented in (Karttunen, 1977) is too complicated to explain in a few words. The following is my own proposal developed in lecture notes. It
computes the same meanings as the analysis of Karttunen and that of (Groenendijk and Stokhof, 1982). Consider the complement clause of the following sentence. Its standard meaning is indicated in the line below.

(85) A whether-question  
Bill wonders whether Mary snores  
Bill wonders \( \lambda w. \lambda p. p = \text{that Mary snores} \lor p = \text{that Mary doesn’t snore} \land p(w) \)

If we want an analysis compatible with what we have already, whether Mary snores should be a wh-phrase that is moved to [Spec, CP]. The wh-phrase must be the existential quantifier \( \lambda w. \lambda P_{(st)t} (\exists p_{st})[p = \text{that Mary snores} \lor p = \text{that Mary doesn’t snore} \land P(p)] \). This reflection motivates the following meaning rule for whether:

(86) \( F(\text{WHETHER}_{(st)((st)t)t}) = \lambda w. \lambda q_{st}. \lambda P_{(st)t}.[P(q) \lor P(\lambda w'. \neg q(w'))] \)

As the type of the operator makes clear, it applies directly to the embedded clause and must be QR-ed such that it has wide scope with respect to the question operator ? . I am assuming that at SS there is as semantically empty operator whether in C or in [Spec, CP], which is licensed by WHETHER under agreement. whether is deleted at LF. The WHETHER-phrase is moved to [Spec, CP] generating an interpretable LF. Here is the derivation:

\[
\text{[CP whether [C'[(st)t]WH-PRO] [(st)t]WHETHER [(st)t] [t Mary snores]]]  
\]

(SS)  

\( \Rightarrow \) (WH-movement, whether-deletion, WH-PRO-movement)  

\( [(st)t] \lambda_{2,at} [(st)t]WHETHER [(st)t] [t Mary snores] [(st)t \lambda_1 [C'[(st)t] t_{2,at} t_{1,at}]] \)

LF

Note that both kinds of WH-movement applied in the last step leave a trace of type st. The reader may convince herself by own computation that the LF expresses the intension indicated in (85). In the analysis given, the vacuous wh-word whether functions as a scope marker. In some languages, e.g. German, scope markers occur in other interrogative constructions as well:

(87) Partial wh-movement  
Was glaubst du was Willi meint welches Mädchen schnarcht?  
what believe you what Bill thinks which girl snores  
\( = \lambda w. \lambda p.(\exists x) [x \text{ is a girl in } w \land p = \lambda w'. \text{you believe in } w' [\lambda w''. \text{Bill thinks in } w'' [\lambda w'''. [x snores in w''']]]]] \)
The standard analysis assumes that the wh-phrase \textit{was}, which occurs in the two higher [Spec,CP]'s, is semantically empty. \textit{Was} functions as a scope marker, indicating that the interpretable wh-phrase \textit{welches Mädchen} ‘which girl’ is moved to the matrix [Spec, CP] along the \textit{was}-path. Constructions like these are called \textit{partial wh-movement}: the interpretable wh-phrase has been moved to the local [Spec, C] at SS but not to the position where it is interpreted at LF.

There are syntactic restrictions on WH-movement. Usually, an interrogative is an island for extractions:

(88) *What$_1$ do you remember who bought t$_1$

\[\lambda w.\lambda p.(\exists x)[x \text{ is something in } w \land p = \text{ that you remember who bought } x \land p(w)]\]

The intension listed under the sentence shows that there is no semantic reason for why the sentence should be bad. If the wh-word in the matrix [Spec,C] originates in the matrix, such an extraction is possible, however. In fact, the following sentence is ambiguous (see (Baker, 1968)):

(89) A \textit{Baker-sentence}

Who remembers where we bought what?

a. \[\lambda w.\lambda p.(\exists x)[\text{person}(w)(x) \land p = \lambda w'.x \text{ remembers in } w' \land \lambda w''.'\lambda q.(\exists y)[\text{place}(w''')(y) \land (\exists z)[\text{thing}(w'''')(z) \land q = \lambda w'''.\text{we bought } z \text{ at } y \text{ in } w''''] \land q(w''')] \land p(w)'\]

b. \[\lambda w.\lambda p.(\exists x)[\text{person}(w)(x) \land (\exists z)[\text{thing}(w)(z) \land p = \lambda w'.x \text{ remembers in } w' \land \lambda w''.'\lambda q.(\exists y)[\text{place}(w''')(y) \land q = \lambda w'''.\text{we bought } z \text{ at } y \text{ in } w''''] \land q(w''')] \land p(w)\]

We obtain the first reading if \textit{what} is moved to the embedded [Spec,CP] at LF. The second reading is obtained if \textit{what} is moved to the matrix [Spec,CP] at LF. In a way the matrix wh-word, i.e., \textit{who} serves as an optional scope marker for the embedded \textit{what}.

Questions exhibit the phenomenon of \textit{Pied-Piping}, which requires reconstruction at LF. Pied-Piping means that a wh-phrase cannot be extracted on syntactic grounds at SS. Therefore it drags quite a bit of surrounding stuff with it when it is WH-moved. The pied-piped material is reconstructed at LF to an interpretable position. Here are two examples with simplified LFs.

(90) a. SS: \[[C[P[AP t$_3$ $\text{how tall}$]]$_1$ $[C[C[i$ ?$] [S Martin$_3$ t$_2$ t$_1$]]]]]$

      LF: \[[how$_4$ $[[? \text{Martin}[t$_4$ $\text{tall}]]]]\]
b. SS: [CP whose dog \(1 \in C \) ?] [s Tom \(t_2 \in t_1\)])

LF: [CP who \(3 ? \) [s Tom is [(t\(3 \) `'s] dog)])]

c. SS: How old a man is Bill?

LF: How \(1 ? \) Bill is a \(t_1 \) old man

d. SS: What an incredible idiot John is!

LF: what \(1 ? \) Bill is a \(t_1 \) incredible idiot

To see that these LFs are reasonable (and in fact required), let us analyse (90a) in some detail. tall is a degree adjective. The standard analysis of degree adjectives takes them as a relation between individuals and degrees. We integrate the new type \(d\) (“degrees”) in our system. For the example given, the relevant degrees are meters (subdivided in centimetres and so on). The semantic domain of \(d\), \(D_d\), contains different sorts of degrees (degrees of length, degrees of mass, and others). We have to choose the right ones for each adjective. I will use the variable \(d\) for referring to degrees as well. The interpretation of the adjective is:

(91) A degree adjective

\[
F(\text{tall}_{d(et)}) = \lambda w. \lambda d. \lambda x. \text{the height of } x \text{ in } w \text{ is at least } d \\
\lambda w. \lambda d. \lambda x. \text{H}(x, w) \geq d
\]

how is an existential quantifier over degrees:

(92) \(F(\text{how}_{dht}) = \lambda w. \lambda P_{dht}(\exists d) P(d)\)

Let us consider the LF in (90a), which is more accurately the following:

(93) \(\lambda s[ \text{how}_{d,t} \ [(? t_5.et)[[\text{Martin} \ t_4 \text{tall}]]]]\)

I leave it to the reader to convince herself that the tree expresses the intension \(\lambda w. \lambda P_{dht} (\exists d)[[p = \lambda w'.\text{H(Martin}, w') \geq d] \& p(w)]\). Suppose Martin is 2.05 m tall in world \(w\). Then this meaning assigns \(w\) the following set of propositions:

\{that Martin is d-tall: \(d \leq 2.05 \text{ m}\}\.

Arguably this is the correct meaning.

To analyse the example in (90b), we need a meaning for the possessive determiner [s], which is present in whose:

(94) \(F([e(s(et))e] \ s) = \lambda w. \lambda x. \lambda P_{s(et)}; \text{ There is a unique } y \text{ such that } P(w)(y) \& x \text{ possesses } y \text{ in } w. \text{ the unique } y \text{ such that } P(w)(y) \& x \text{ possesses } y \text{ in } w\)

The clause after the colon is the presupposition of the determiner, written in H&K’s notation. We return to presuppositions below. Suppose Tom is the dog of Mary in world \(w\).
Then the LF in (90b) assigns w the singleton set \{that Tom is Mary’s dog\}.

The analysis of questions given raises the question of how the reconstruction process assumed for the derivation of the LFs works. Let us reconsider (90b) again. The SS of the complex DP could be something like this:

(95) \[Dep [Det' [DP who] [Det s]] [NP dog] \]

We have to move the generalised quantifier who for scope reasons. Extraction out of a determiner (left branch extraction by the terminology of J.R. Ross) is not possible. Therefore who pied-pipes the entire DP. This produces the SS in (90a), which is not interpretable. If we disregard the movement of the finite verb is and we assume Internal Merge in the style of MP, the full spell-out of the SS is the following:

(96) \[CP [Dep [Det' [DP who] [Det s]] [NP dog] ] [c[c ? WH-PRO] [s Tom is [Dep [Det' [DP who] [Det s]] [NP dog]]]]\]

If we try to obtain an interpretable LF by an appropriate deletion, we see that we are not entirely successful. We delete the pied-piped material of the DP in [Spec, CP] and leave it in the base position:

(97) \[CP [Dep [Det' [DP who] [Det s]] [NP dog] ] [c[c ? WH-PRO] [s Tom is [Dep [Det' [DP who] [Det s]] [NP dog]]]]\]

But that doesn’t help. We cannot interpret the index of the base DP as a bound variable as the semantics of questions requires. What we need is a variable 1 of type e in the place of the embedded who. So we seem to need a special indexing convention for WH-movement. The following one has the desired effect:

(98) If a WH-phrase is moved by Internal Merge, the WH-phrases proper are co-indexed and the pied-piped material remains outside of the index.

Thus, the SS of (90a) is not (96), but rather:

(99) \[CP [Dep [Det' [DP who] [Det s]] [NP dog] ] [c[c ? WH-PRO] [s Tom is [Dep [Det' [DP who] [Det s]] [NP dog]]]]\]

Now we can apply FI and wh-PRO-moving and obtain the desired LF. After deletion of the uninterpretable categories and Operator-movement, we obtain the fully interpretable LF:

(100) \[w-PRO2 [CP [Dep [Det' [DP who] [Det s]] [NP dog]] [c[c ? WH-PRO2] [s Tom is [Dep [Det' [DP who] [Det s]] [NP dog]]]]\]
\[ \lambda_{2,x}[\text{who} \lambda_{1,e} [\text{Tom is } [[[t_1]'s] \text{ dog}]]] \]

The deletion process plus the co-indexing convention thus seem to circumvent long extraction at LF. Yet the avoidance is only apparent. It presupposes an ad hoc co-indexation of the two wh-phrases proper. A procedure that would first extract who of a wh-phrase that has pied-piped material and is located in [Spec, CP] and would reconstruct the pied-piped stuff to the base position would be equivalent. I proposed such a procedure in (von Stechow, 1996).

Here is a summary of the results of this section.

1. The construal of LFs for questions requires WH-movement at LF, i.e. covert movement. This holds for multiple questions and for whether-questions.

2. The construal of the LFs require WH-PRO-movement thus providing additional evidence that PRO-movement is a device available to grammar.

3. WH-movement might be rather long violating island constraints. In this case, the target of the movement is often indicated by a surface device such as a WH-phrase in the higher clause or by a scope marker (Germ. was).

4. The phenomenon of pied-piping requires reconstruction of a rather complicated kind. If we realise it by deletion, we need a special convention for co-indexing. We co-index the WH-phrases proper, i.e. the pied-piped material is left outside.

All these principles apply on the LF branch.

9.3. Notes on the literature

The classics in the semantics of questions are (Hamblin, 1973), (Karttunen, 1977) and (Groenendijk and Stokhof, 1982). The syntax for questions in Karttunen’s original is more complicated; a lot is done by syncategorematic rules. The LF for constituent questions used here has emerged over the years in Irene Heim’s lecture notes and publications; see, e.g. (Heim, 1994b). The analysis for WHETHER and the introduction of wh-PRO are my own additions. Different approaches to partial wh-movement are found in (von Stechow et al., 2000). There are attempts to interpret questions by means of choice functions, Skolem functions and a combination thereof; cf. (Engdahl, 1980, Engdahl, 1986) and (Reinhart, 1994). The most thorough discussion of these approaches is found in unpublished lecture notes of Irene Heim: (Heim, 1994e). A lot of this material has been used in (von Stechow, 2000).
10. COMPARATIVE AND SUPERLATIVE

Comparative constructions encompass the comparative proper, the superlative and degree quantifiers like most or few. They show that we have to move degree operators for scope reasons in the syntax and at LF. Monotone decreasing quantifiers like few students or less than 3 students will contain a quantifier-like negative element that has to be QR-ed at LF. Many comparative constructions are elliptic, which forces us to speak about the interpretation of ellipsis and its position in the syntax-semantics interface.

10.1. Comparative: QR and Ellipsis Interpretation

We start with the following comparative construction.

(101) The table is longer than the drawer is wide.

The standard assumption is that the than-clause is the complement of the comparative -er. The example is one of the few cases where the complement clause is a full CP. But in most cases, the than-clause is reduced:

(102) a. Bill is taller than Mary is tall
    b. Bill is taller than Mary is tall

In (a) the VP is deleted (“Comparative Deletion”) and in (b) only the AP is deleted (“Comparative Subdeletion”); cf. (Bresnan, 1973).

The following examples show that the than-clause is extraposed to the right periphery of a sentence:

(103) John was climbing a higher tree than John was.

(104) Nicole made more money last year than Tom did.

Recently it has been claimed by (Bhatt, 2002) that the extraposition is actually due to QR of the DegP whose head is the comparative morpheme -er. Let us see what is behind such a claim. Ignoring tense, the standard analysis of the comparative -er is this (cf. (Heim, 2001) among others):

(105) The comparative operator ER: category Deg, type (dt)((dt)t)

\[
\text{F(ER)} = \lambda w. \lambda P_d. \lambda Q_d. (\forall d)[P(d) \rightarrow Q(d)] \land (\exists d)[Q(d) \land \neg P(d)]
\]

\[
= \lambda w. P \subset Q
\]
ER is thus simply a strong universal quantifier over degrees, which says that its restriction is a proper subset of its nuclear scope. (The restriction of a quantifier Det is its first argument; the nuclear scope is its second argument.)

We know that generalised quantifiers must be QR-ed when they occur in object position. Here is the derivation of the LF for the sentence in (101):

\[(106) \quad [S \text{the table}_1 [VP \text{is } [AP t_1 [\text{DegP},(dt)t \text{ ER [CPWH}_2 \text{than the drawer is } t_2 \text{ wide]] long}_{d(et)}])]\]

This structure assumes WH-movement in the than-clause to generate a property of degrees. Furthermore, the subject is raised out of the AP. The structure exhibits the problem of the object because the DegP has type (dt)t, but the adjective long requires an argument of type d. We resolve the type conflict by QR-ing the DegP. Suppose now that the DegP is not adjoined to the left periphery of the sentence but to the right one. This generates the following SS:

\[(107) \quad [S[S[\text{the table}_1 [VP \text{is } [AP t_{1,e} t_{3,d} \text{ long}_{d(et)}])] [\text{DegP},(dt)t \text{ ER [CPWH}_2 \text{than the drawer is } t_2 \text{ wide]]}]]\]

After deletion of the uninterpretable stuff by FI, we obtain a fully transparent LF:

\[(108) \quad [([\lambda_3[\text{the table}\lambda_1 [ t_{1,e} t_{3,d} \text{ long}_{d(et)}]]) [\text{DegP},(dt)t \text{ ER [CPWH}_2 \text{than the drawer is } t_2 \text{ wide]}])]
\]

\[= \lambda w.\lambda d[\text{the drawer in } w \text{ is } d\text{-wide in } w] \subset \lambda d[\text{the table in } w \text{ is } d\text{-long in } w]\]

Suppose the structure in (107) is the input for PF. Then the extraposition of the than-clause is an example of an application of QR at SS.

There is a problem for theories of this kind, however. The comparative morpheme -er is not pronounced at its SS-position. It is a part of the inflection of the adjective long-er. If ER were identical with the inflection -er, it would be necessary to reconstruct it into the verb at PF. Then the verb must contain an -er trace. Recall, however, that ER is a sort of determiner. I don’t know of any syntactic process that could move a part of the adjective to the quantifier position. Therefore, I will assume that ER is a phonetically empty operator that has the feature [iER], while an adjective with comparative morphology has the feature [uER]. The interpretable feature licenses the uninterpretable feature under binding. So the official representation is this:

\[(109) \quad [S[\text{the table}_1 [VP \text{is } [AP t_1 [\text{DegP},(dt)t \text{ ER[iER]} [CPWH}_2 \text{than the drawer is } t_2 \text{ wide]] long-er}_{d(et)}[u-more]]])\]

After QR-ing the DegP to the right periphery we obtain:
The feature transmission proceeds like this: the interpretable feature origins with the semantic operator ER. It is transmitted under binding as [uER] to its trace t₃. The trace agrees with the [uER] feature of the comparative adjective longer and licences its morphology. The comparative adjective longerd(et) therefore means exactly the same as the adjective stem long, because the inflection -er has no meaning. This move makes (110) a good PF candidate. After PF-deletion of ER and the other covert operators, everything is at the place where it is pronounced.

This split between an invisible operator that is interpreted and a visible morphology that is not interpreted might seem artificial. We will see however, that other phenomena of natural language must be interpreted along the same lines, e.g. tense, aspect and n-words. We will discuss these in sections 11, 12, and 14. I will assume therefore that the move is justified.

A note on the semantics of antonyms is in order. Recall that the meaning of “positive” degree adjectives was given in (91), i.e. long has the meaning \( \lambda w. \lambda d. \lambda x. \text{LENGTH}_w(x) \geq d \). Antonyms are negative in the sense that they are obtained from positive adjectives via internal negation (cf. (Heim, 2001)):

(111) \[
\text{Antonyms} \quad \[ \text{short}_d \] = \lambda w. \lambda d. \lambda x. \neg \text{LENGTH}_w(x) \geq d \quad (\text{i.e. } \text{LENGTH}_w(x) < d)
\]

The reader may convince herself that we can analyse a sentence like The rope is shorter than the snake is in an adequate way using this semantics.

Let us take up the question what the problem of ellipsis resolution in comparative constructions tells us about the SS/LF interface. Consider the following sentence:

(112) Caroline observed more birds than Arnim did [VP \Delta]

If we assume that DegP has been QR-ed at SS-structure already, the only thing that remains to do is to fill the VP-gap \( \Delta \) via copying:

1. \( \lambda_1 [\text{Caroline} [\text{VP observed} t_1 \text{ more birds}]] [\text{ER than Arnim} \text{ did} [\text{VP} \Delta]] \) \quad \text{SS}
2. \( \lambda_1 [\text{Caroline} [\text{VP observed} t_1 \text{ more birds}]] [\text{ER than Arnim} \text{ did} [\text{VP observed} t_1 \text{ more birds}]] \) \quad \text{VP-copying}

We have to QR the two DPs [t more birds] in object position and we have to apply FI and obtain a fully interpretable LF, which expresses the following proposition:
The comparative adjective *more* means the same as *many*. Recall that the comparative morphology means nothing, but it points to the semantic operator \(ER\). We assume that the plural DP in object position has a covert indefinite article, which denotes the existential quantifier for pluralities.

\[
(114) \quad F(\text{more}_{d(et)}) = F(\text{many}_{d(et)}) = \lambda w. \lambda d. \lambda x. |x| \geq d, \text{ where } |x| \text{ denotes the cardinality of } x 
\]

\[
(115) \quad F(\exists_{et}(\text{et})) = \lambda w. \lambda P_{et}. \lambda Q_{et}. P \cap Q \neq \emptyset
\]

The precise structure of the object DP in the example is therefore:

\[
(116) \quad [\text{DP}, \text{et}][\text{NP}, \text{et}][\text{AP}, \text{et} t_1 \text{more}][\{\text{et} \text{birds}\}]
= \lambda w. \lambda P_{et}. (\exists x)[x \text{ are birds in } w & |x| \geq d & P(x)]
\]

(The trace \(t_1\), i.e., the free variable \(d\) is bound by \(ER\), which is located higher up in the LF.) We will say later how plural predication works. If we take it for granted, this is precisely the meaning we need for the construction.

Comparative deletion as exhibited by example (102a) is analysed exactly alike. The same holds for the subdeletion in (102b); the only difference is that this time the gap is an AP in the *than*-clause. I leave it to the reader to work out the ellipsis resolution for that example.

### 10.2. Quantifiers in the *than*-clause

The analysis given in this section cannot explain the behaviour or quantifiers in *than*-clauses. Consider the sentence:

\[
(117) \quad \text{Mary is taller than every boy}
\]

If the scope of *every boy* is the complement clause, the sentence should mean: “Mary is taller than the shortest of the boys”. This reading is not attested. So matrix scope of *every boy* is obligatory and one would like to know why this should be so. Let us call the former reading the Min-reading and the latter one the Max-reading. Here is a representation of the truth-conditions:

\[
(118) \quad \begin{align*}
\text{a. } & \lambda w. \lambda d. (\forall x)[\text{boy}(x,w) \rightarrow H(x,w) \geq d] \subset \lambda d[H(Mary,w) \geq d] \quad \text{(Min)} \\
\text{b. } & \emptyset \lambda w. (\forall x)[\text{boy}(x,w) \rightarrow \lambda d[H(x,w) \geq d] \subset \lambda d[H(Mary,w) \geq d]] \quad \text{(Max)}
\end{align*}
\]

The intuitively correct Max-reading is expressed in (118b). It requires QR of *every boy* out
of the complement clause and thus violates Clause Boundedness.

Perhaps type lifting can overcome the problem. (Heim, 2001, Heim, 2003) proposes two type-lifts that obtain the correct reading, apparently without long QR.

The first step is to type-lift the comparative operator ER. Note first that the most natural interpretation of ER is the relation > that holds of two degrees d and d’ if d > d’. This is the simplest possible meaning of the comparative and the operator should have the type d(dt). (The version used in section 10.1 had the type of a quantifier, i.e. (dt)((dt)t) and was already lifted so to speak.) The lifted type that is the analogue to the type of a lifted transitive verb is ((dt)t)((dt)t,t).⁷ Heim’s entry for the lifted ER-operator is this:

\[(119) F(ER^* ((dt)t)(((dt)t),t)) = \lambda w.\lambda P_{dt}.\lambda Q_{dt}.Q(\lambda d(P(\lambda d'(d > d'))))\]

The next step of Heim is to introduce an operator that lifts sets of degrees to generalised quantifiers of degrees:

\[(120) Heims PI-operator^{8}\]

PI is of type (dt)((dt)t).

\[F(PI) = \lambda w.\lambda D_{dt}.\lambda D'_{dt}.\lambda \text{Max}(D') \in D, \text{ where Max}(D') \text{ is the greatest degree in } D'.\]

With these two operators we can represent the Max-reading of the sentence in (118) without involving long QR:

\[(121) Mary \text{ is taller than every boy}\]

---

⁷ A lifted transitive verb takes two quantifiers as arguments. Take for instance the verb like, which has the type e(et). The lifted version like* has the type ((et)t)(((et)t)t). Its meaning would be \(\lambda w.\lambda Q_1.\lambda Q_2.Q_2(\lambda x.Q_1(\lambda y.x \text{ likes } y \text{ in } w))\).

⁸ “PI” stands for “points to intervals”.
By several applications of $\text{FA}$ and $\lambda$-conversion, we calculate that this means:

$$\lambda w.\text{ER}(\lambda D. (\forall x)[\text{boy}(x, w) \rightarrow H_w(x) \in D])(\lambda D. H_w(\text{Mary}) \in D)$$

$$= \lambda w. (\forall x)[\text{boy}(x, w) \rightarrow H_w(\text{Mary}) > H_w(x)]$$

This is the correct reading and every boy remains within the complement CP. We still have to QR every boy over the PI-operator and thus don’t get rid of QR. Furthermore we have to block an LF where every boy has narrow scope with respect to the PI-operator, because that would give us the Min-reading. Some boy requires wide scope with respect to the PI-operator as well because the sentence Mary is taller than some boy doesn’t have the Max-reading. On the other hand, the negative polarity item (NPI) any boy must have narrow scope with respect to the PI-operator, because the sentence Mary is taller than any boy only has the Max-reading. Thus we have to derive somehow the following distributional pattern:

(122) a. Mary is taller than every boy (every boy > PI, *PI > every boy)
    b. Mary is taller than some boy (some boy > PI, *PI > some boy)
    c. Mary is taller than any boy (PI > any boy, *PI > any boy)

(> stands for “wide scope with respect to”). It is unclear how these restrictions can be derived. Heim’s treatment of the comparative therefore helps to get rid of long QR, but it cannot eliminate the need for QR.

Another phenomenon that speaks against the elimination of QR is provided by DegP-movement of superlatives for the relative/comparative reading of the sentence in (123), which will be discussed in the next section. The sentence will require movement of the superlative operator to an adjunction site of the VP.

We conclude from this discussion that the Clause Boundedness Restriction is basically correct, and type lifting cannot eliminate QR.

Here is the summary of the results obtained in this section.

1. The comparative operator $ER$ is a covert quantifier, which licenses the comparative morphology of the adjective under agreement.

2. The DegP is QR-ed in the syntax carrying along the elliptic than-clause. This is the SS that serves as an input for the PF-rules.

3. The VP- or AP-ellipsis is resolved by copying another VP or AP located in the same sentence. This method is known as “$\Delta$-interpretation”.

4. After copying, we still need the rule of WH- insertion.
5. Quantifiers in the *than*-clause that are not NPIs must take scope over the comparative operator *ER*. If we want to avoid long QR, we need Heim’s PI-operator and a lifted version of *ER*. We still need the stipulation that the quantifiers in question are scoped over PI.

10.3. Superlative

Superlative adjectives contain a superlative operator *EST* that has to be moved for type reasons. The movement occurs at the LF branch and perhaps even in the syntax. The analysis assumed here is due to (Heim, 2004 (1999)). The following sentence has two readings, called the *absolute* and the *comparative* one, where the latter is sometimes called *relative* as well.

(123) John climbed the highest mountain.

a. John climbed the highest of the mountains. (absolute)

b. John climbed a higher mountain than Bill and Mary did. (comparative)

The standard of comparison that is responsible for the different readings is provided by a domain variable *C* whose value is determined by a context assignment *g*. Suppose *C* contains the mountains *a, b, c*, among which *a* is the highest. Then the truth condition for the absolute reading is this:

(124) John climbed a mountain that is higher than any other mountain.

To get this, we need a slightly more complicated meaning for degree adjectives: they have the same semantics as before but intersect with the head noun.

(125) \[F(tall_{(et)(d(et))}) = \lambda w.\lambda P_{et}.\lambda d.\lambda x. H_w(x) \geq d \land P(x)\]

Thus, the NP *[d tall mountain]* is true of an individual if it is a mountain that is tall to at least degree *d*. The superlative operator has the following meaning:

(126) The superlative *EST*

has the category Deg and the type (*et)*((d(et))(et))

---

9 This meaning rule is due to (Heim, 2004 (1999)), who follows Seuren’s comparative semantics. A rule more in the style of the comparative semantics used in this article would be this (with presuppositions ignored):

\[F(EST) = \lambda w.\ldots [\exists d.\lambda x. (\forall y \in C) [y \neq x \rightarrow \lambda d. R(d)(y) \subset \lambda d. R(d)(x)]]\]
\[ F(EST) = \lambda w. \lambda C_{et}. \lambda R_{(d(et))}. \lambda x \in C. (\exists d)[R(d)(x) \land (\forall y \in C)[y \neq x \rightarrow \neg R(d)(y)]] \]

EST is covert and licenses the superlative morphology -est under agreement.

SS and LF for the absolute reading are these:

(127) SS:  John climbed [the [NP [AP EST highest] mountain]]

LF:  John climbed [the [NP [NP,et ESTC [d(et) \lambda_d [NP,et][AP d highest] mountain]]]]

The LF is derived by QR-ing EST to the NP hosting it. The domain variable C is the first argument of EST. It is written as an index to facilitate reading. Suppose now that C has the value \{Mont Blanc (a), Monte Rosa (b), Matterhorn (c)\}. Suppose now that John climbed the Mont Blanc. Then the LF in (127) expresses the following truth-value:

(128) John climbed in w that x \in C such that (\exists d)[x is a d-high mountain in w \land (\forall y \in C)[y \neq x \rightarrow y is a mountain in w that is less than d-high]]

This is the truth because among the three the Mont Blanc is the highest.

A note on the types is in order. In previous paragraphs, degree adjectives had the simpler type d(et) and were combined with the head noun by means of PM (after saturation of the degree argument). Now the adjectives are classified as NP-modifiers, i.e. as attributes. After the degree argument is plugged in, the adjective has the type (et)(et). The adjective is combined with the NP via FA then. We could have used the simpler type et as well, but that would overgenerate. Suppose high is of type et and ESTC is interpreted in situ. Then [ESTC highest] would express the property \( \lambda x. (\exists d)[x is d-high \land (\forall y \in C)[y \neq x \rightarrow y is less than d-high]] \). Consider the DP the ESTC high building under the assumption that C contains the town hall (a), the Empire State Building (b), Mount Washington (c) and Mount McKinley (d). Then the Empire State Building should be the highest building with respect to C, but the semantics would not yield that, because b is not the highest thing with respect to C. But among the buildings in C, b is the highest. So C should only contain buildings, not things in general. This motivates the slightly more complicated type of the adjective.

The comparative reading requires that ESTC is QR-ed up to the VP:

(129) John ESTC \lambda_d [climbed [THE [NP[d highest] mountain]]]

If we try to interpret this LF, we notice that the definite article makes no sense here for the LF would mean: “John climbed the mountain that has a certain height d and everyone else didn’t climb that mountain”. This can be true if John climbed a particular mountain and
somebody else climbed a higher mountain. The sentence cannot mean that.

Already (Szabolcsi, 1986) noticed that the definite article can have an existential interpretation in relational (comparative) superlative constructions, i.e., it expresses the existential quantifier. We therefore establish the following convention: THE can mean $\exists$ if the definite interpretation makes no sense. If we assume this, the quantifier in (129) is an existential quantifier and has to be QR-ed inside the VP, using the PRO-method:

$\lambda d [\exists \lambda_2 [\lambda_1 \lambda_2 [\text{THE} (= \exists) [\text{NP} [\text{d highest mountain}] \lambda_2 [t_1 \text{climbed } t_2]]]]$

$= \lambda w. (\exists d) [\lambda_1 \lambda_2 [\lambda_1 \lambda_2 [\text{THE} (= \exists) [\text{NP} [\text{d highest mountain}] \lambda_2 [t_1 \text{climbed } t_2]]]]$

Arguably, this is a correct interpretation of our sentence.

If we embed the sentence under a modal we encounter the familiar de re/de dicto ambiguity, which is due to the circumstance that the object may have wide or narrow scope with respect the modal. The ambiguity is multiplied because in addition to the object the superlative operator may occur at different positions:

(131) John has to climb the highest mountain.

a. absolute de dicto

$\lambda_1 \lambda_2 [\lambda_1 \lambda_2 [\text{THE} (= \exists) [\text{NP} [\text{d highest mountain}] \lambda_2 [t_1 \text{climbed } t_2]]]]$

b. comparative de dicto

$\lambda_1 \lambda_2 [\lambda_1 \lambda_2 [\text{THE} (= \exists) [\text{NP} [\text{d highest mountain}] \lambda_2 [t_1 \text{climbed } t_2]]]]$

c. absolute de re

$\lambda_1 \lambda_2 [\lambda_1 \lambda_2 [\text{THE} (= \exists) [\text{NP} [\text{d highest mountain}] \lambda_2 [t_1 \text{climbed } t_2]]]]$

d. comparative de re

$\lambda_1 \lambda_2 [\lambda_1 \lambda_2 [\text{THE} (= \exists) [\text{NP} [\text{d highest mountain}] \lambda_2 [t_1 \text{climbed } t_2]]]]$

I leave it to the reader to imagine different scenarios for the different LFs. The comparative de re requires unorthodox application of QR. We QR the object and the subject. In the next step we QR ESTC to a position between the $\lambda$-abstract created by the movement of the subject and the subject. This is possible under H&K’s assumption that the movement index is a $\lambda$-operator that is attached to the landing site.

There have been attempts in the literature to interpret the domain variable $C$ in situ and interpret the superlative without DegP-movement. For the examples discussed so far this is perhaps possible. (Heim, 2004 (1999)) has shown, however, that there is a further reading that cannot be obtained without non-local DegP-movement, at least not by standard methods. The
reading presupposes a scenario like this: John has to climb a 5000 m high mountain (in order to qualify), Bill has to climb a 4000 m high mountain and Mary has to climb a 3000 m high mountain. The LF that is true in this scenario is this:

(132) The fifth reading

\[
\text{John} \, \text{EST}_C \, \lambda d \, \lambda t_1 \, \text{has} \, t_1 \, \text{to climb THE} \, d\text{-highest mountain}
\]

\[C = \{\text{John, Mary, Bill}\}\]

Again, this LF requires the unfamiliar application of QR, which has been mentioned. (Szabolcsi, 1986) observes similar examples.\(^{10}\)

### 10.4. “Most” and “Fewest”

The quantifier **most** is the most important piece of the theory of generalised quantifiers. The reason is that its meaning cannot be expressed in first order logic but requires higher order quantification. The standard analysis since (Barwise and Cooper, 1981) is something like (a) or (b) in the following definition:

(133) A proportional quantifier

\[
\text{a. } F(\text{most}) = \lambda w. \lambda P, \lambda Q. |P \cap Q| > |P - Q|
\]

\[
\text{b. } F(\text{most}) = \lambda w. \lambda P, \lambda Q. |P \cap Q| > |Q|/2
\]

In other words, **most** Ps are Qs means: “There are more P & Qs than P & non-Qs” or: “The number of the P & Qs is bigger than the number of half of the Qs”. In (Hackl, 2000) and (Hackl, 2006) it has been shown that this analysis is not general enough:

1. It cannot explain certain ambiguities we know from the superlative operator.
2. It cannot explain why the quantifier **fewest** cannot have a proportional reading.
3. The analysis doesn’t relate the quantifier to the superlative, but it obviously is a superlative.

Comparison with German shows that **most** is not a quantifier at all but a superlative adjective:

(134)\(^{a}\) **Hans hat die meisten Berge bestiegen.**

Hans has the most mountains climbed

b. Hans climbed more mountains than anybody else. \[(\text{relative})\]

\[^{10}\text{E.g. her (27a) “Who expected to get the fewest letters?”}, \text{which has the reading ‘Who expected to get fewer letters than anybody else expected to get?’}.\]
c. Hans climbed more than half of the mountains.  (proportional)

The sentence has the two readings indicated. The relative reading (Hackl’s term) is the one we called comparative reading in the previous section. The proportional reading is related to the absolute reading of the last section but it is not the same. The generalised quantifier most can only derive the proportional reading.

The absolute reading noted in the previous section would mean “all of the mountains”. Already (Szabolcsi, 1986) noticed that the absolute reading doesn’t exist. Another problem the generalised quantifier account cannot explain is the meaning of fewest:

(135)  
a. **Hans hat die wenigsten Berge bestiegen.**  
Hans climbed the fewest mountains

b. Hans climbed fewer mountains than anybody else. (relative)

c. *Hans climbed less than half of the mountains.  (proportional)

The analysis of the relative and the proportional reading of most is straightforward. The consequences of the analysis of superlative constructions are similar to those for comparative constructions. We first reformulate the lexical entry for many in (114) to make the adjective attributive:

(136) **many** revised:

\[ F(\text{many}_{(et)(d(et))}) = \lambda w. \lambda P_{et}. \lambda d. \lambda x. P(x) \land |x| \geq d \]

The LFs (modulo QR of the object) of the relative and the proportional reading in (134) are then the following ones:

(137) The relative reading

**Hans EST\textsubscript{C} \lambda d. climbed [THE d many mountains]**

\[ = \lambda w. (\exists d)(\exists x)[\text{mountains}_{w}(x) \land |x| \geq d \& \text{Hans climbed } x \text{ in } w \]

\[ \& (\forall y)[y \in C \land y \neq \text{Hans } \rightarrow \neg(\exists x)[\text{mountains}_{w}(x) \land |x| \geq d \& y \text{ climbed } x \text{ in } w]] \]

Recall that the morphological realisation of many is most and that THE is the existential quantifier. If we assume that \( C = \{\text{Hans, Bill, Mary}\} \), the LF means that Hans climbed more mountains than Bill and Mary.

(138) The proportional reading

**Hans climbed [THE EST\textsubscript{C} \lambda d.[d many mountains]]**

\[ = \lambda w. (\exists d)(\exists x)[\text{mountains}_{w}(x) \land |x| \geq d \& (\forall y)[y \in C \land y \neq x \rightarrow \neg|y| \geq d \& \text{mountains}_{w}(y)] \land \text{Hans climbed } x \text{ in } w] \]
Two assumptions are necessary to understand the reading. (i) Pluralities are distinct (≠) only if they don’t overlap. (ii) C contains at least two distinct objects, here mountains. (iii) Hans climbed a group x if he climbed each member of x. Assumption (ii) accounts for the lack of the absolute reading. Suppose that C contains five individual mountains m1,…,m5 and the sums of them. To see the adequacy of the truth condition, suppose Hans climbed m1+m2+m3. The mountains distinct from this group are m4+m5, m4 and m5. Each of these groups has a smaller cardinality than the group climbed by Hans.

Hackl claims that this account gives us an explanation of why fewest cannot have a proportional reading. Suppose few has the following semantics:

$\text{(139) } F(\text{few}(\text{et})(\text{d(et)})) = \lambda w.\lambda P.\lambda d.\lambda x. P(x) \& |x| \leq d.$

(Another possibility would be to have |x| < d in the definition.) The proportional reading would then be represented as:

$\text{(140) Hans climbed [THE EST}_C \lambda d.\{d \text{ few mountains}\]}$

$= \lambda w. (\exists d)(\exists x)[\text{mountains}_w(x) \& |x| \leq d \& (\forall y)[y \in C \& y \neq x \rightarrow \neg |y| \leq d \& \text{mountains}_w(y)] \& \text{Hans climbed } x \text{ in } w]$ 

This truth condition can never be satisfied. Suppose we have three mountains m1, m2, m3. Which is a smallest set of mountains? It can’t be m1, because |m1| is not smaller than |m2| or |m3|. A similar reasoning holds of m2 and m3. So there is no smallest set.

The derivation of the relative reading requires a special device. The straightforward representation in the style of (138) with few instead of many is inconsistent for similar reasons as (140). The derivation of the relative reading will be addressed in the next section.

Here is the summary of the discussion:

1. The superlative EST operator is covert and licenses the superlative morphology under agreement.

2. The superlative operator is a DegP with a type that requires FA to an argument of type d(et), which has to be created by movement. The adjunction sites are typically an NP with a degree adjective as attribute or a VP.

3. The domain variable, which is to be filled by the context, is crucial for the different readings. The LF-position of ESTC partially determines the value of C.

4. The movement required may have an unfamiliar landing site between a λ-operator created by previous movement and the antecedent of that λ-operator. We will find the same kind of configuration in plural constructions.
10.5. Comparative Quantifiers

Let us take up the problem of representing the comparative reading for sentence (135). As I said, the straightforward analysis parallel to (137) cannot be correct:

\[(141) \quad \textit{Hans EST}_C \lambda d. \text{climbed [THE d few mountains]}\]

\[\lambda w. (\exists d)(\exists x)[\text{mountains}_w(x) \land |x| \leq d \land \text{Hans climbed } x \text{ in } w \land (\forall y)[y \in C \land y \neq \text{Hans} \rightarrow \neg(\exists x)[\text{mountains}_w(x) \land |x| \leq d \land y \text{ climbed } x \text{ in } w]]\]

Suppose Hans climbed one mountain and the others each climbed two mountains. Then each of the others also climbed one mountain. So this analysis is inconsistent. The problem is in fact more general and arises with any monotone decreasing quantifier: we cannot represent those as existential quantifiers without some additional operations, though that works with monotone increasing quantifiers. This fact is known as van Benthem’s problem (cf. (van Benthem, 1989)). Consider the following minimal pair:

\[(142) \quad \begin{align*}
\text{a. Many students passed.} \\
\text{b. Few students passed.}
\end{align*}\]

Let us assume the positive operator of (von Stechow, 2007):

\[(143) \quad \text{The Positive-operator} \]

\[\text{has the type } (dt)t. \ F(\text{POS}_{N_c}) = \lambda w. \lambda D_{dt}. (\forall d \in N_c) \ D(d).\]

\(N_c\) is a contextually given “neutral interval” of the relevant degree scale. For instance, the degrees that count as neither few nor many. The left boundary of \(N_c\) marks the border line of the “few” domain and the right boundary of \(N_c\) marks the border line to the “many” domain. Similarly for adjectives like short/small and other antonym pairs. Sentences with monotone increasing quantifiers are then represented in the way to be expected:

\[(144) \quad \text{POS } \lambda d \ [\exists \ d\text{-many students}] \text{ passed} = \lambda w. \ (\forall d \in \text{POS}_w)(\exists x)[|x| \geq d \land \text{students}_w(x) \land \text{passed}_w(x)]\]

\(\exists\) is the familiar existential quantifier. POS is QR-ed at LF for type reasons.

A similar analysis for (142b) yields a trivial reading:

\[(145) \quad \text{POS } \lambda d. \ [\exists \ d\text{-few students}] \text{ passed} = \lambda w. \ (\forall d \in \text{POS}_w)(\exists x)[|x| < d \land \text{students}_w(x) \land \text{passed}_w(x)]\]

Suppose that the admissible threshold for being few students is the number 4. Then the sentence “Twenty students passed” should be false. But the analysis predicts it to be true.
because of the monotony condition contained in the adjective \textit{few}. Perhaps this reading is unacceptable for pragmatic reasons, i.e. we silently add the Gricean exhaustifier meaning “and no more than few”. For other comparative quantifiers such as “the fewest”, “fewer than three” or “at most three” the exhaustification seems belong to the meaning. So let us assume that this is generally the case for monotone decreasing quantifiers.

A possible way of implementing the readings in question is Heim’s (2006a) decomposition of \textit{few} into a type-lifted negation + MANY\textsuperscript{11}. MANY is covert. \textit{few} is type-lifted negation of properties of degrees that has to be extracted from the DP for type reasons:

\begin{equation}
\text{few is of type } \text{d}((\text{dt})\text{t}). F(\text{few}) = \lambda w. \lambda d. \lambda D_{dt}. \neg D(d)
\end{equation}

The analysis of (142b) is now this:

\begin{equation}
\text{DS: } \exists \text{ [POS few] MANY students] passed} \\
\Rightarrow 2 \times \text{QR} \\
\text{LF: POS } \lambda d'. \ [d' \text{ few}] \lambda d \ [\exists d-\text{MANY students} ] \text{ passed} \\
= \lambda w. (\forall d \in N_c)(\exists x)[|x| \geq d \& \text{students}_w(x) \& \text{passed}_w(x)]
\end{equation}

This is the reading we want.

Applying this decomposition, we can derive the relative reading for Hackl’s sentence:

\begin{equation}
\text{DS: Hans climbed [THE EST}_C \text{ fewest MANY mountains]} \\
\Rightarrow 4 \times \text{QR} \\
\text{LF: Hans EST}_C \lambda d'. \lambda y. [d' \text{ fewest } \lambda d[[\text{THE d MANY mountains} ] \lambda x. y \text{ climbed } x]] \\
= \lambda w. (\exists d)(\exists x)[|x| \geq d \& \text{mountains}_w(x) \& \text{Hans climbed}_w(x)] \& \\
(\forall y \in C)[y \neq \text{Hans} \rightarrow (\exists x)[|x| \geq d \& \text{mountains}_w(x) \& \text{y climbed}_w(x)]]
\end{equation}

Suppose Hans climbed one mountain, Bill climbed two, and Mary climbed three mountains. Take d = 2. Then Hans didn’t climb two mountains, but Bill and Mary did. So this account is correct.

A similar analysis applies for decreasing comparative DPs with extraposed \textit{than}-clause:

\begin{equation}
\text{Arnim observed fewer birds than Caroline did.}
\end{equation}

\textsuperscript{11} The article quoted treats \textit{little} in this way. But the idea is the same.
\[ \lambda d. \text{Arnim observed} d\text{-}\text{few} \lambda d' \exists d'\text{-MANY birds} \]
\[ \exists d'\text{-MANY birds} \]
\[ = \lambda w. \lambda d. \neg (\exists x)[d\text{-}\text{many birds}_w(x) \& \text{Arnim observed}_w x] \supset \]
\[ \lambda d. \neg (\exists x)[d\text{-}\text{many birds}_w(x) \& \text{Caroline observed}_w x] \supset \]

The LF ignores the LF movement of the objects. Comparative quantifier like **more than 3** and **less than 3** have a somewhat lexicalised analysis. The *than*-clause should mean something like “3 are many”, but it can never be overt nor is it extraposed.

(150) More than 3 students passed.

**LF:** \([\text{more than 3}] \lambda d. \exists [d \text{ MANY students}] \text{ passed} \]
\[ \lambda w. \lambda d. \exists [\text{students}_w(x) \& |x| \geq d \& \text{passed}_w(x)] \supset \lambda d'. 3 \geq d' \]

Again, the lexical entry of more enforces the application of QR at LF:

(151) \( more than n \)

\[ F(\text{more}_{d(d(t),t)}) = \lambda w. \lambda d. \lambda D_{d(t)} D \supset \lambda d'. d \geq d' \]

As to be expected, the lexical entry for *less* contains a negation, which corresponds to the meaning of Heim’s *few*:

(152) \( Less than n \)

\[ F(\text{less}_{d(d(t),t)}) = \lambda w. \lambda d. \lambda D_{d(t)} \lambda d'. \neg D(d') \supset \lambda d'. \neg 3 \geq d'. \]

(153) Less than 3 students passed.

**LF:** \([\text{less than 3}] \lambda d. \exists [d \text{ MANY students}] \text{ passed} \]
\[ \lambda d. \neg \exists [\text{students}(x) \& |x| \geq d \& \text{passed}(x)] \supset \lambda d. \neg 3 \geq d \]

According to (Geurts and Nouwen, 2006) the quantifiers **at least/most 3** contain an epistemic component:

(154) I have at least 3 children.

\[ = \text{It is certain that I have 3 children and it is possible that I have more than 3 children.} \]

The sentence is odd because it suggests that the speaker ignores the precise number of her children. Similarly:

(155) I have at most 3 children.

\[ = \text{It is possible that I have 3 children and it is not possible that I have more than 3 children.} \]
It could be that the epistemic effect belongs in effect to the semantics of the quantifier. I will ignore it here and merely analyse the comparative aspect. Sentences of our simple kind have the following truth-conditions then:

(156) At least 3 students passed.
\[ \lambda.d.\exists x[\text{students}(x) & |x| \geq d & \text{passed}(x)] \supseteq \lambda.d.3 \geq d. \]

(157) At most 3 students passed.
\[ \lambda.d.\neg \exists x[\text{students}(x) & |x| \geq d & \text{passed}(x)] \supseteq \lambda.d.3 < d. \]

The following two meaning rules derive these readings after applying long QR at LF:

(158) \textit{at least} \ n
\[ F(\text{at least}_{d(dt,t)}) = \lambda.w.\lambda.d.\lambda.D_{dt}.\lambda.d'.D(d') \supseteq \lambda.d'.d \geq d' \]

(159) \textit{at most} \ n
\[ F(\text{at most}_{d(dt,t)}) = \lambda.w.\lambda.d.\lambda.D_{dt}.\lambda.d'.\neg D(d') \supseteq \lambda.d'.\neg d \geq d' \]

The account of comparative quantifiers given in this section uniformly assumes an indefinite article in determiner position. This is different from the analysis found in Generalised Quantifier theory, which assumes a multiplicity of determiners. Furthermore, all comparative quantifiers are reduced to the semantics of the comparative or the equative. The monotone decreasing quantifiers all contain a negation in their meaning that takes wide scope with respect to the existential component at LF making the quantifiers negative. (This explains why they license a negative polarity item in their scope; NPIs are not treated in this article.)

The analysis assumes that we are able to QR across DP violating the left-branch constraint. If we scope the negation bearing elements, e.g. Heim’s few, inside the DP, we generate the trivial van Benthem reading. As an example consider an illicit LF for few students:

(160) DS: \[ \exists [\text{NP PRO} [\text{POS} \text{few} [\text{MANY students}]]] \]

\[ \text{LF: } \exists [\text{et PRO}_1 [\text{POS}_3 [t_3 \text{few } t_1 t_2 \text{et } t_1 [\text{det MANY students}]]]] \]

\[ = \lambda.w.\lambda.P.(\exists x)(\forall d \in N_c)\neg [|x| \geq d & \text{students}_w(x) & \text{passed}_w(x) & P(x)] \]

Summary of this section. The analysis of decreasing comparative quantifiers requires degree operators that contain negation of properties of degrees. These must be QR-ed for type reasons at LF. They out-scope the DP. So these constructions require an LF-movement that violates the left-branch constraint.
10.6. Notes on the literature

Most approaches to the syntax of comparative constructions follow the seminal article (Bresnan, 1973). The semantics for ER and that of degree adjectives follows (Heim, 2001). Pioneer articles on degree semantics are (Cresswell, 1976) and (von Stechow, 1984). The method of dealing with ellipsis via copying (“\(\Delta\)-interpretation”) is due to (Williams, 1977). (Schwarzschild and Wilkinson, 2002) try to explain the scope behaviour of quantifiers in than-clauses; (Heim, 2001) and (Heim, 2006b) explain their method as a lifting of the comparative complement to a higher type in the way presented in section 10.2.

Pioneering work to the superlative are (Heim, 1985) and (Szabolcsi, 1986). The crucial example that requires a movement analysis of the superlative operator is due to (Heim, 2004 (1999)) as far as I know. Recent work on comparative and superlative quantifiers is (Hackl, 2000, Hackl, 2006). (Sharvit and Stateva, 2002) try to derive comparative/relative readings for superlatives without non-local movement of EST by type lifting devices.

The classical reference to the semantics of generalized quantifiers is (Barwise and Cooper, 1981). Much of the literature following that work focuses on logical properties of generalised quantifiers neglecting issues in syntax and compositionality. Compositional approaches to comparative quantifiers in the style presented here build on Hackl’s work. Domain variables for quantifiers have been introduced into the literature by (von Fintel, 1994).

11. TENSE

Tense and aspect are marked at the verbal morphology, but the semantic tense or aspect operator is not interpreted at the verb. Semantic tenses are covert operators that license their morphology under agreement. Thus the familiar picture of covert operators reappears. The treatment of tense requires more complicated types for natural language predicates. In the previous sections, predicates were relations of type \(e^n t\), i.e. the types consisted of a sequence of \(e\)’s and ended on \(t\). Now the last argument has to be the type \(i\), which stands for times, i.e., the predicates will have the type \(e^ni\). In the section treating aspect, a further refinement will be introduced. We will assume that the set of types is increased by a new type \(i\) (time intervals) and the functional derivatives thereof: \(D_i = T\), \(T\) the set of all time intervals. Times are partially ordered by the relation \(<\) “before” and \(>\) “after”. Time intervals are coherent sets of time points. Hence they may overlap, they may stand in the inclusion relation and so on.
11.1. Simple Tenses and Temporal PPs

Verbs and other predicates have a temporal argument of type i. We assume that the argument is the last one. This is conventional. The time argument could have been the first equally well. Up to now we have assumed that verbs like \textit{called} have the type et. If we include tense, the verb has type e(it). The entry of the verb is this:

(161) \textit{Past tense morphology}

\[ F(\text{called}_{e(it)}) = \lambda w.\lambda x.\lambda t.x \text{ calls in } w \text{ at time } t. \]

The verb has the feature \{uP\} “uninterpretable Past”

It is important to keep in mind that \textit{called} has a tenseless meaning. It means precisely the same as \textit{calls}. This is so because the tense morphology has no semantic interpretation. But the morphology points, of course, to a semantic tense located elsewhere in the clause.

(162) \textit{Semantic Past} is of type i(it,t).

\[ F(\text{P}_i(it,t)) = \lambda w.\lambda t.\lambda P_i.t \exists t'(t' < t \& P(t')) \]

\textit{P} has the feature \{iP\} “interpretable Past”

The first argument of \textit{P} is a time. If this time is supplied by the semantic Present, we have a \textit{deictic Past}. If the time is supplied by another semantic Past, we have a Pluperfect.

(163) \textit{Semantic Present}

\[ F_c(N_i) = t_c, \text{ where } t_c \text{ is the speech time; } N \text{ has the feature \{iN\} “interpretable Present”} \]

(The letter “N” reminds of “now”.) Here is the analysis of a Simple Past sentence

(164) Mary called.

\[ [\text{TP PAST}(N) \text{ PRO}_1 [\text{VP} t_1 \text{ Mary called}]] \]

\begin{center}
\begin{tabular}{c c c}
\hline
\textit{[iP]} & \textit{[uP]} & \textit{[uP]} \\
\hline
\end{tabular}
\end{center}

\[ = \lambda w.(\exists t < t_c) \text{Mary calls in } w \text{ at } t. \]

TP stands for tense phrase. \textit{PRO} is Heim & Kratzer’s empty pronoun, which creates a \textit{\lambda} operator because the type of the complex tense operator \textit{P(N)} requires an argument of type it. \textit{PRO}_1 is the semantic binder of the time variable \textit{t}_1. We assume Heim & Kratzer’s theory of semantic binding according to which the expression \textit{\alpha} that causes a bound variable by abstraction counts as a derived semantic binder. Thus the operator \textit{P(N)} binds the variable \textit{t}_1. It transmits its feature \{iP\} as the uninterpreted variant \{uP\} to \textit{t}_1. The \{uP\} or \textit{t}_1 agrees
with the [uP] feature of the finite verb, thus licensing its morphology. Feature transmission by binding is indicated by ___. Feature agreement is indicated by ….

Let us consider a present statement next. Like verbs adjectives have a temporal argument, but adjectives don’t have temporal morphology and therefore cannot realise tense at the surface. The realisation is done by the auxiliary be, which has a very trivial semantics, viz. identity:

\[(165) \text{The temporal auxiliary } be \]
\[F(is/was_{it}) = \lambda w.\lambda t.P(w.t)\]

To give an example, consider the LF of Mary is sick:

\[(166) [TP N PRO_2 [VP t_2 \text{ is PRO}_1 [AP t_1 \text{ Mary sick}]]] \]
\[[iN] \quad [uN] \quad \lambda w.\text{Mary is sick in } w \ \text{at } t_c\]

Given that the Present N has type i, we could have generated it at the position t_2 without the application of PRO-movement. I stick to this representation in order to stress the parallelism with the former representation.

The following sentence shows that the semantic past can be quite distant from the morphological past:

\[(167) \text{Last week Mary called every day.}\]

Ignoring the perfective aspect of the VP (which is treated in section 12.1), the meaning of the sentence is something like this:

\[(168) \lambda w.(\exists t_2)[t_2 < t_c & t_2 \text{ is in the last week} & (\forall t_1)[t_1 \text{ is a day} & t_1 \text{ is on } t_2 \rightarrow \text{Mary calls at } t_1 \text{ in } w]]\]

The LF could be something like this:

\[(169) [TP [P(N)] PRO_3 [VP[PP \exists \text{ IN THE last week IN } t_3] PRO_2 [VP[PP \exists \text{ ON every day IN } t_2] PRO_1 [VP t_1 \text{ Mary called}]]]]\]

The SS contains a cascade of temporal PPs of which the higher may be regarded as a frame adverbial because its object is a definite time span that serves as a restriction for the embedded quantifier. The PPs contain some unpronounced but interpreted material viz. the existential temporal quantifier \(\exists_{(it)(it)}\), the prepositions IN/ON, which both denote the
subinterval relation \( \subseteq \), and the definite article THE. The LF shows that the semantic tense is quite distant from the morphological tense and determines the verbal morphology via feature transmission under semantic binding.

In (von Stechow, 2002), I generated the LF in (169) by an iterative application of QR. I started from nested PPs that got their surface order by QR. For instance, the source of the two temporal adverbs \textit{last week} + \textit{every day} was something like \([ \text{PP} \exists \text{ON} \text{every day IN} \text{PP} \exists \text{IN THE last week IN [PAST\text{t}]} ]\). Here, QR is replaced by appropriate PRO-movement. Temporal PPs are quantifiers of type \((\text{it})t\). We create the type of the subordinate quantifier by appropriate PRO-movement. Thus the order of the adverbials at DS is the same as that at LF. Semantically, both methods come to the same.

Summary of this section: Verbs have a time argument that is filled with a variable that has to be semantically bound. The binders by default are the semantic tenses Present (N) and Past (P). P has a time variable as well. The operator expresses a relative variable with respect to the argument time. We obtain a deictic reading, if the argument is Present. The semantic tenses have an interpretable feature ([iN]/[iP]). This is transmitted under semantic binding to the bound variables. A finite verb has the feature [uN] or [uP]. This feature has to agree with the u-feature of the verb’s temporal variable. Temporal adverbs are temporal quantifiers of type \((\text{it})t\). The type of their nuclear scope is generated by PRO-movement, where PRO originates in a time position of type i. PRO movement is used for temporal binding quite generally.

11.2. A note to the Passive

In English, \textit{be} has other meanings as well. In section 12.2 we will see that it can express the Progressive. Another function is that it may express the Passive. (Chomsky, 1981) says of the Passive that it absorbs objective case and blocks the subject theta-role. Therefore the object has to be raised to the subject position for case reasons. We obtain this effect if the passive auxiliary is an existential quantifier:

\[
\text{F(is/was}_{e(it)(in)}) = \lambda w. \lambda P_{e(it)}. \lambda t.(\exists x)P(x)(t)
\]

The auxiliaries have the feature [iPASS] and [uN]/[uP]

Here is the analysis of \textit{Mary is loved}:

\[
\text{DS: } \text{N [PRO [is [loved Mary]]]}
\]

\[
\text{SS: Mary }\lambda_1 \text{N PRO}_2 [\_ [t_2 \text{[is [loved t_1]]}]]
\]
The passive participle has the same meaning as the transitive verb **love**. The feature [uPASS] is responsible for case absorption. **Mary** is raised to the subject position for case reasons.

We know that semantic subjects can be introduced by **by**-phrases. **by** has a very trivial meaning. It expresses a sort of identity.

(172) Passive **by**

\[ F(\text{by}(e(it)), (e(it))) = \lambda w.\lambda x.\lambda P(e(it)).\lambda y.\lambda t. x = y & P(y)(t) \]

The important point is that the **by**-phrase has to be adjoined to the passive participle phrase before the existential closure achieved by the passive auxiliary:

( 173) Mary is loved by everyone.

DS:  N [PRO [is [[loved Mary] [by everyone]]]]

LF:  **Mary** \( \lambda_1 \text{everyone} \lambda_3 \text{N PRO}_2 [t_2 \text{is } [e(it)] [\text{loved } t_1] [\text{by } t_3]] \)

\[ \lambda w.(\forall x) [\text{person}(x,w) \rightarrow (\exists y) [x = y & y \text{ loves Mary in w at } t_c]] \]

\[ = \lambda w.(\forall x) [\text{person}(x,w) \rightarrow x \text{ loves Mary in w at } t_c] \]

I leave it to the reader to convince herself that this implementation of Chomsky’s passive theory is compatible with more complicated raising constructions like the following one:

(174) Mary is believed to be loved by everyone.

11.3. Complex Tenses

The complex tenses are formed by the auxiliaries **have** and **will**. Both auxiliaries are time shifters: **have** shifts the evaluation time to a previous time and **will** shifts it to a later time.

(175) The **Perfect auxiliary** (to be refined)

\[ F(\text{has/had}(e(it)), (e(it))) = \lambda w.\lambda P(e(it)).\lambda t. (\exists t' < t)P(t') \]

These auxiliaries are semantic tenses, but they have no i-feature. The present form has the feature [uN] and the past form has the feature [uP].

Here is the analysis of a Pluperfect sentence:

(176) John had called.

\[ [\text{TP} P(N) \text{ PRO}_1 [\text{VP} t_1 \text{ had } \text{PRO}_2 [\text{partP} t_2 \text{John called}]]] \]
Note that the feature [uP] is passed to the participle. The latter is a tenseless form and therefore doesn’t need a tense feature at its variable. We will see that it is useful to have it there, however.

For German, we can safely assume that hat (‘has’) and hatte (‘had’) have the semantics indicated in (175). For English, a complication arises. In most cases the present form has expresses a so-called Extended-Now Perfect.

\[
F(\text{has}) = \lambda w. \lambda P_u \lambda t. (\exists t')[XN(t', t) \& P(t')],
\]

were $XN(t', t)$ is true iff $t$ is a proper final subinterval of $t'$.

Consider the analysis of a present perfect sentence under this analysis:

(178) Mary has been sick.

\[
\begin{align*}
N & \text{PRO1} [VP t_1 \text{has PRO2} [t_2 \text{been PRO3} [t_3 \text{Mary sick}]]] \\
& \lambda w. (\exists t_2)[XN(t_2, t_c) \& \text{Mary is sick in } w \text{ at } t_2]
\end{align*}
\]

The analysis assumes that been is the past participle of the temporal auxiliary be, which has been introduced in (165). The reading entails that Mary is still sick at the speech time. The XN-Perfect can explain what (Klein, 1992) calls the Present Perfect Puzzle, i.e. the incompatibility of the present perfect with temporal adverbs that give us a past time span such as yesterday:

(179) \[F(\text{yesterday}(it)(it)) = \lambda w. \lambda P_u \lambda t. (\exists t')[t \subseteq \text{the day before the day that contains } t_c \& P(t)]\]

The XN-Perfect introduces a time span that includes the speech time. This time span cannot be a part of yesterday. Hence the oddness of the example in (180) is explained:

(180) *Mary has called yesterday.

The Future is expressed by the auxiliary will, which is the mirror operator of the Perfect.

(181) \textit{The Future auxiliary will}

\[\begin{align*}
F(\text{will}(it)(it)) & = \lambda w. \lambda P_u \lambda t. (\exists t')[t' > t \& P(t')]
\end{align*}\]

will has the feature [uN].

Thus the sentence Mary will call has the analysis:

(182) \[N \text{PRO1} [t_1 \text{will PRO2} [t_2 \text{Mary call}]]\]
There is no indicative form in the past for will. One might wonder why this is so. Under the present account this does not follow from anything.

Summary of this section. The auxiliaries has/had and will are time shifters. They act like semantic tenses but have no i-feature. They transmit the u-feature of their variable ([uN] for has/will, [uP] for had. For the Syntax/Phonology interface the present system entails that PRO-movement must occur at the surface, because it is necessary both for the semantic interpretation and for feature transmission. The phonetic spell-out of tensed forms requires that the u-features have been checked.

11.4. Sequence of Tense

The analysis given can explain that subordinate tense formes might be semantically tenseless. The phenomenon is addressed as Sequence of Tense (SOT). The sentence

(183) John believed Mary was sick. (simultaneous)

\[
\begin{array}{c}
\text{P(N) PRO}_1 [t_1 \text{John believed PRO}_2 [t_2 \text{ was PRO}_3 [t_3 \text{ Mary sick}]]] \\
[iP] [uP] \ldots [uP] [uP] \ldots [uP]
\end{array}
\]

\[
\lambda w. (\exists t > t_c)(\forall w', t')(w', t') \in \text{Dox}_\text{John}(w, t) \rightarrow \text{Mary sick}(w')(t')
\]

has a “simultaneous” reading. Under this reading John worded his belief about Mary as: “Mary is sick”. We obtain this meaning if we interpret the object clause as tenseless. This is possible, if the semantic tense in the matrix clause checks both the morphological past of believed and that of was under binding. The distribution of the tense feature is indicated in (183). This analysis makes it possible that there is only one semantic tense in the construction. The meaning for the verb believed has to be this:

(184) \text{Propositional attitudes}

\[
\begin{array}{c}
F(\text{believed}_{(s(it))(e(it))}) = \lambda w.\lambda P_{s(it)}.\lambda x.\lambda t.(\forall w', t') \in \text{Dox}_x(w, t): P(w')(t'),
\end{array}
\]

where Dox$_x(w, t) =$ the world-times compatible with what x believes in w at time t.

(Note that the composition of this verb with its complement CP requires IFA.) The backward shifted reading of the sentence in (183) is obtained by embedding a semantic Past in the complement. This operator introduces a new binding chain:

(185) John believed Mary was sick. (shifted)

The time argument of the embedded P is bound by the matrix P. But the embedded Past has an inherent i-feature and thus opens a new binding chain. So the time variable $t_3$ of was is bound by the lower Past. Note incidentally that the construction shows the usefulness of the decomposition of semantic Past into N+P. If Past were always a deictic tense, the analysis of the backward shifted reading would not have been possible.

The following sentence shows that features are transmitted through non-finite forms:

(186) John will say that Mary is sick.

There is no semantic Present in the complement. In fact, the complement is tenseless and interpreted “simultaneous” with the saying in the future. The tense feature of the embedded is is licensed by the matrix Present, which transmits its feature through the finite form will and through the non-finite form say under binding.

An interesting prediction of the account is that Present under Past is not possible in these constructions.

(187) #John believed that Mary is sick.

The only way to license the present morphology in the complement is to bind the time variable of is by the matrix Present. This, however, delivers the wrong logical type for the complement, which has to be of type it rather than being of type t. Sentences of this type are not plainly ungrammatical. They express the so-called Double Access reading, which requires a different syntax; cf. (Abusch, 1993), (Ogihara, 1995), (Kratzer, 1998) for relevant proposals.
Summary of this section. Verbs of propositional attitudes count as temporal binders. They transmit the u-feature of their time variable to the highest bound time variable in the subordinate clause. This account entails the distribution of tenses in SOT-constructions.

11.5. Tense in Relative Clauses

(Partee, 1973) gives arguments that tenses behave like pronouns. In view of the behaviour of Past in shifted contexts this cannot be true of Tense in general, but in relative clauses this seems to be so. We assume that the time variable of the main verb in a relative clause is realized as a temporal pronoun $\text{Tpro}$ that has to be bound by a higher tense, which is a stipulation. Let us discuss how this works and consider a sentence taken from (Kusumoto, 1999):

(188) Mary talked to a man who was crying like a baby.

The sentence has three readings: (a) the crying can be simultaneous to the talking; (b) it can be before the talking; (c) it can be after the talking. The representations of these are the following ones:

(189) Simultaneous Reading

$\text{N PRO}_1 \text{P}(t_1) \text{ PRO}_2 [\text{t}_2 \text{ Mary talked-to} [\text{a man} [\text{WH}_3 [\text{Tpro}_2 \text{ PRO}_4 [\text{t}_4 \text{ was PRO}_5 [\text{t}_5 \text{ crying}]]]]]]]$

$iP$ $[uP]$ $[uP]$ $[uP]$

Here $\text{Tpro}_2$ is bound by the matrix Past. It is precisely the simultaneous reading that forces us to assume a $\text{Tpro}$ in relative clauses. Unlike for tenses under propositional attitudes, the verb talked-to is not a binder of the time variable $t_4$ in the relative clause. The latter is not a complement of the verb but an adjunct to a noun.

(190) Shifted Reading

$\text{N PRO}_1 \text{P}(t_1) \text{ PRO}_2 ... [\text{WH}_3 [\text{Tpro}_2 \text{ PRO}_4 \text{P}(t_4) \text{ PRO}_5 [\text{t}_5 \text{ was PRO}_6 [\text{t}_6 \text{ t}_5 \text{ crying}]]]]]$

$\text{iP}$ $[uP]$ $[uP]$ $[uP]$

$\text{iP}$ $\text{uP}$

The difference between the two structures is that the second structure has a semantic Past under $\text{Tpro}_2$. This licenses the tense morphology and is responsible for the backwards shift. Still, the time variable $t_4$ of the embedded Past is bound by the matrix Past. The independent reading is obtained by binding the $\text{Tpro}$ to the matrix Present while having a relative Past in its scope:

(191) Independent Reading

$\text{N PRO}_1 \text{P}(t_1) \text{ PRO}_2 ... [\text{WH}_3 [\text{Tpro}_1 \text{ PRO}_4 \text{P}(t_4) \text{ PRO}_5 [\text{t}_5 \text{ was PRO}_6 [\text{t}_6 \text{ t}_5 \text{ crying}]]]]]$

69
Relative clauses may occur with nouns that are embedded under attitudes. The account given here applies to such cases as well.

11.6. Notes on the literature

The classic in tense semantics is (Prior, 1967). Prior’s theory assumes that propositions are sets of times, i.e., they have the type it. Tenses are propositional operators. The Present is nothing, P is a backwards shifter (P = \( \lambda p. \lambda t.(\exists t' < t)p(t') \)), and the future F is the corresponding forward shifter.\(^{12}\) The main defect of this account is that there are no temporal variables in the syntax that can be bound. Many of the examples treated in the previous section required the presence of temporal variables. Much of the literature following Prior motivated the need of temporal variables in the syntax: (Bäuerle, 1979), (Cresswell, 1990) among others.

The system presented here evolved over the years. The idea that temporal features are transmitted by semantic binding is due to Irene Heim: cf. (Heim, 1994d), (Heim, 2005). The present spell out of the idea for temporal binding is due to Heim as well (personal communication). That tense under attitudes is vacuous and gets its features somehow via binding is an idea found in (Kratzer, 1998) though the semantics of tense is referential there and the mechanism of feature transmission is different. Kratzer assumes a referential theory of tense, inspired by (Partee, 1973). Partee’s argument for the referential character of tenses like the Past comes from negation. The sentence in (192a) means neither (192b) nor (192c).

(192) a. I didn’t turn off the stove.
    b. \( \lambda w. \neg (\exists t < t_c) I \text{ turn off the stove in } w \text{ at } t \)
    c. \( \lambda w. (\exists t < t_c) \neg I \text{ turn off the stove in } w \text{ at } t \)

As (Heim, 1997) has pointed out, the sentence might very well mean something like (192c) provided we assume that existential quantifiers come into life with a variable that restricts the domain of quantification. In other words, our operator P takes a set of times as the first argument and means this:

(193) Past with domain restriction

\(^{12}\) These rules are attributed to Prior by everyone, but I have not been able to find them in this form in his work.
\[
F(P_{\text{it}(\text{it},\text{it}))} = \lambda C \cdot \lambda t \cdot \lambda P_{\text{it}}(\exists t')(C(t') \land t' < t \land P(t'))
\]

At LF, C is a free variable, whose value is determined by the assignment function. Suppose C is a set of times the speaker has in mind. Then the LF for (192a) is the following and means that the speaker didn’t turn off the stove at a particular past time she has in mind.

\([t_2 \textbf{I turn off the stove}]\]

The observation that tense under attitudes must be ignored in the semantics has a long tradition as well. Here are a few references: (Ogihara, 1989) works with tense deletion; (Ogihara, 1995) does SOT with an embedded relative Present that is identity; (Abusch, 1993) has a very complicated mechanism of tense licensing. (von Stechow, 1995) has a system of bound tense in subordinate constructions that comes close to binding by PRO-movement; the system is inspired by (Heim, 1994a). The system presented here is very close to that in (Kusumoto, 1999); the treatment of tense in relative clause is almost identical with her proposal. A theory of bound tense is developed in several paper by Schlenker, starting with his dissertation (Schlenker, 1999).

There is a great amount of literature on tense in English. One of the eternal questions of this language is the treatment of the present perfect. The so-called Extended Now Perfect has its origin in (McCoard, 1978); the formal implementation of the idea is due to (Dowty, 1979). A formal analysis of the German Perfect is found in (Musan, 2002); the book contains a detailed investigation of temporal adverbs and temporal clauses as well. The organisation in terms of i/u-features is inspired by Zeijlstra’s (2004) work on Negative Concord. The idea that features are transmitted by binding is due to Irene Heim.

This article is not concerned with tense in discourse. A standard reference for this topic is (Kamp and Reyle, 1993). The framework used there is a different one.

The last remark concerns the Passive auxiliary. Despite the abundant literature on the Passive, there seems to be very little about the semantic interpretation of the Passive construction. I have analysed the passive auxiliary in the way indicated in lecture notes and talks. (Di Sciullo and Williams, 1987) account of the Passive and the by-phrase might be compatible with the analysis proposed here.

12. **Aspect**

Languages differ with respect to aspect marking. English only marks the Progressive by a special syntax/morphology. In the Slavic languages verbs come in pairs, i.e., they have a perfective and an imperfective variant. This section treats the Perfective and the
Progressive. Aspects exhibit the same systematic behaviour as tense: they are marked in the verbal morphology but they do not affect the verb meaning directly. Rather they point to a covert semantic operator that applies to the VP. Semantically, aspects are operators that map properties of events into properties of times. Henceforth we will assume that we have the type v (events) in our system: \( D_v = V = \) the set of events.

12.1. The Perfective

We start with Slavic aspect and consider Russian. Russian verbs have two morphological forms: the imperfective and the perfective one. For instance, the form “read” has the variants chital (imperfective) and prochital (perfective). The entry for the perfective form is this:

(195) A perfective verb

\[
F(\text{prochitala}_e(e(vt))) = w.\lambda x.\lambda y.\lambda e.y \text{ reads } x \text{ in } (\text{the event }) e \text{ in } w
\]

The verb has the features \([uPF], [uP], [uFemale]\)

(I will neglect gender agreement here.) PF stands for PERFECTIVE (not to be confused with PERFECT!), whose semantics is this:

(196) The Perfective operator PF

\[
F(\text{PF}_e(v)(it)) = w.\lambda P.\lambda t. (\exists e)[\tau(e) \subseteq t \& P(e)],
\]

where \(\tau(e)\) is the running time of the event \(e\).

This semantics is a formalization of the account in (Klein, 1994). Consider now the analysis of the sentence:

(197) Masha prochitala pis’mo.

“Mary did a complete reading of the letter”

The SS is the following:

(198) \( N \ PRO_3 [\text{TP } P(t_3) \ PRO_2 [\text{AspP } t_2 \ PF \ PRO_1 [\text{VP } t_1 \text{ Masha prochitala [THE pis’mo]]}]]\)

\[
\begin{array}{cccc}
\hline \hline \\
\end{array}
\]

\[
[iPF] \quad [uPF] \quad [uPF] \quad [uPF] \\
\hline \hline \\
\]

\[
= w.(\exists t < t_i)(\exists e)[\tau(e) \subseteq t \& \text{Mary reads the letter in } e \text{ in } w]
\]

pis’mo means ‘letter’. Since Russian doesn’t express the definite article, we have to assume a covert one. Once the feature mechanism is understood, the analysis is very simple and
Two remarks are in order. The first concerns the entries of verbs. In the last section we said that the type of predicates terminate on the functional type it. This is not true anymore for verbs. They terminate on vt. We have labour sharing between the aspectual operator and the semantic tense. The former converts a set of events into a set of times. The latter converts a set of times into a truth-value. The second remark concerns feature transmission. The verb has the features u-Past and u-Perfective. The latter comes from the Perfective operator PF as we would expect. The former stems from the Past operator P and is transmitted by binding. Now PF seems to open a new binding chain, because it has the inherent feature [iPF] and event variables are of a type different from that of time variables. Time features and aspect features are, however, not in conflict with each other. The feature [uP] is transmitted through the operator PF(t2) and we have one big binding chain here. On the other hand, the feature [uN], which stems from the semantic Present is not passed through P(t3) because P has the inherent feature [iP], which is incompatible with [uN].

Since this treatment might be unfamiliar to the reader, I give the surface structure for the sentence:

(199) Masha prochitala pis’mo.

English has no perfective morphology, but it has been argued in the literature that we may
have a covert perfective operator nevertheless (cf. (Klein, 1994)). If we accept this, a more accurate LF for the example in (164) would be something like the following structure:

\[(200) \quad [t \text{ N PRO}_3 \text{P}(t_3) \text{ PRO}_2 \{ t_2 \text{ PF PRO}_1 \{ t_1 \text{ Mary called} } ] ] ]
\]

\[= \lambda w.(\exists t)[t < t_c & \exists e[\tau(e) \subseteq t & \text{Mary calls in } e \text{ in } w]]
\]

Since *called* doesn’t have the feature [uPF], it needs not be checked in the syntax. Nevertheless we have to assume the Perfective operator for type reasons, provided the verb has the type e(vt).

### 12.2. The Progressive

The English progressive morphology is encoded by the participle ending –*ing*, which has the feature [uPROG]. The Progressive operator is the auxiliary *be*, which must not be conflated with the temporal auxiliary *be*. The semantics for the Progressive is due to (Dowty, 1979: chap. 3).

\[(201) \quad \text{The Progressive auxiliary } \text{be}
\]

\[F(\text{is/was}_v(v_t)((v_t))) = \lambda w.\lambda P(v_t). \lambda w.(\forall w')[w' \in \text{Inert}(w,t) \rightarrow (t')][t \text{ is a non-final subinterval of } t' \& (e)[\exists e[\tau(e) \subseteq t' \& P(w')(e)])],
\]

where \(\text{Inert}(w,t) = \{w': w' \text{ has a common past with } w \text{ up to time } t \text{ and has a future that is expected in view of the facts in } w \text{ at time } t\}.

Thus the Progressive auxiliary is a rather complicated universal modality. It is combined with the VP-argument (the prejacent) by IFA. Inert(w,t) are called “inertia worlds”. As to be expected, *is* has the features [iPROG, uN] and *was* has the features [iPROG, uP].

Consider the analysis of the sentence:

\[(202) \quad \text{Mary was reading a letter.}
\]

\[\text{N PRO}_1 \text{P}(t_1) \text{ PRO}_2 \{ t_2 \text{ was PRO}_3 \{ t_3 \text{ Mary reading a letter} } ] ] ]
\]

\[= \lambda w.(\exists t)(\forall w')[w' \in \text{Inert}(w,t) \rightarrow (t')][t \text{ is a non-final subinterval of } t' \& \exists e[\tau(e) \subseteq t' \& \text{Mary reads a letter in } e \text{ in } w']]
\]

(The variables \(t_2\) and \(t_3\) and their binders could have been left away, and the types would still be correct. We have chosen this LF to make the binding chains transparent.) The
meaning of the verb *reading* is, of course, the same as that of *reads* or *read*.

Summary this section. Formally, aspectual operators look very similar to tenses. Tenses map sets of times to sets of times (or to a truth value), aspects map sets of events to sets of times. Aspectual operators are always in the scope of semantic tense for type reasons. The aspectual operators might be covert like the Perfective in Slavic or they might be overt like the progressive auxiliary *be* in English. Tense features are not in conflict with aspectual features and are therefore passed under binding to the argument of the verb.

12.3. Notes on the literature

There is an abundant literature on Slavic aspect but very little on its semantic interpretation as far as I know. A systematic discussion of different approaches to Russian aspect is found in (Grønn, 2003). As I said the semantics for the Perfective operator formalizes (Klein, 1994). The organisation of the Tense/Aspect system in terms of i/u-features goes back to my lecture notes.

A remark on the Imperfective aspect is in order. It is tempting to assume a semantic Imperfective. The operator could mean that the reference time is in the event time, i.e. the meaning of IPF would be \( \lambda t. \lambda P. (\exists e)[t \subseteq \tau(e) \& P(t)] \). An operator of this kind has been proposed by many authors (e.g. (Bennett and Partee, 1978)), who proposed it as an interpretation of the Progressive. The semantics entails that the event in question comes to an end after the reference time. This is Dowty’s (1979) reason for rejecting the proposal. I think that (Forsyth, 1970) and (Comrie, 1976) are correct in claiming that “the Imperfective” has no uniform meaning in Slavic but a number of different uses: progressive, habitual, general factive (cf. (Grønn, 2003)), two way imperfective and perhaps others. It seems impossible to subsume these under a common meaning. So it is expected that there are other aspectual operators than just the Perfective or the Progressive.

Dowty’s semantics for the Progressive has been occasionally criticised (e.g. by (Landman, 1992)). The criticism is that the accessibility relation Inert sometimes gives the wrong results. But semanticists agree that the basic approach is sound; the accessibility relation might vary with the context.

13. Plural

The semantic plural consists of cumulation operators PL for nouns and * for verbs that cumulate predicates and, more generally, n-place relations. For the LF construal it is important to know that plural operators can be inserted in the syntax at the LF branch if the
semantics requires that. The construction of LFs that represent so-called co-distributive readings require the restriction of the \( \ast \)-operator by a “cover” variable \( C \) and the unconventional parallel QR we have seen in previous sections.

### 13.1. Mereology as a unified ontology of mass and count nouns

We follow (Link, 1991) and (Krifka, 1991) in providing a unified treatment of mass terms and plural terms. In order to do that, we have to say something about the ontology of the domain of individuals \( E \): it must be closed under mereological fusion. For count nouns, each such fusion is a plurality of these things. Think of \( E \) as made of one biggest thing \( U \) and all parts of it. If we write the part-of-relation as \( \leq \) we may define:

(203) The generalised domain of individuals

\[
E = \{ x : x \leq U \}
\]

\( U \) may be thought of as Spinoza’s nature. There might be things that have no proper parts, and those would be the absolute atoms. Most treatments of the plural assume atoms, and we seem to need them for the treatment of count nouns. The NP **six boys** should be true of a plurality \( x \) if \( x \) are boys and \( x \) is a plurality consisting of six atoms. Strictly speaking, this makes no sense, however. If the plurality \( x \) that makes up the six boys consisted of atoms, the particular boys should have no parts. But they have innumerably many. Consider the following sentence, for example:

(204) The boys scratched their noses.

Each boy has a nose as a part, legs, lungs and so on. Each nose is made up of many other parts, and this seems to be true of virtually everything language speaks about. It seems then that the notion of absolute atomicity makes no sense for the semantics of natural language. We need a notion of relative atomicity: \( x \) is a boy-atom if no proper part of \( x \) is a boy. (The proper part relation will be written as \( < \).) Different pluralities will require different relative atoms. This makes the proper treatment of plurality somewhat tedious and might be the reason why most treatments of the plural are sloppy in this respect.

Let us assume then that there are no absolute atoms. We introduce the following terminology for elements in \( E \).

(205) a. \( x \) and \( y \) overlap iff they have some common part:

\[
x o y \text{ iff } \exists z [z \leq x \& z \leq y]
\]

b. \( x \) and \( y \) are distinct iff they do not overlap

(206) Let \( M \) be a subset of \( E \).
\( \Sigma M \) is the fusion of the elements of \( M \) if it has all of them as parts and has no part that is distinct from each of them:
\[
\Sigma M = \text{that } x \in D_c, (\forall y \in M)[y \leq x \& \neg (\exists z)[z \leq x \& (\forall u \in M)[z \text{ is distinct from } u]]
\]
If \( M \) is the finite set \{\(x_1, \ldots, x_n\}\), we write \( \Sigma M \) as \(x_1+\ldots+x_n\).

(207) Let \( P \) be a one-place predicate.

a. \( x \) is a \( P \)-atom in world \( w \) iff \( P(w)(x) \& \neg (\exists y)[y < x \& P(w)(y)] \)
b. \( P \) is quantized if \( P \) is true only of \( P \)-atoms, i.e. \( (\forall w,x)[P(w)(x) \rightarrow x \text{ is a } P \text{-atom in } w] \)
c. \( x \) is a \( P \)-plurality in \( w \) iff \( x \) is a fusion of at least two \( P \)-atoms in \( w \), i.e. \( (\exists M \subseteq P(w))[|M| \geq 2 \& x = \Sigma M] \). If \( y \) is a \( P \)-atom of \( x \), \( y \) is called a member of \( x \), written as \( y \in x \).

Singular count nouns are quantized: no proper part of a boy is a boy. Mass nouns are not quantized, but divisive, every part of milk (or at least many parts of milk) is milk again. Quantized predicates allow us to form pluralities by means of a Plural operator for nouns:

(208) The Plural Operator (for nouns)
\[
[ [ \text{PL} ]] = \lambda w. \lambda P_{(s)} : \lambda x : P \text{ is quantized. } x \text{ is a } P \text{-plurality in } w.
\]

The Plural operator is a distributor: \( \text{PL boy} \) is true of a boy-plurality \( x \) iff every part of \( x \) that is a boy is a boy. Let us say something on the count/mass noun restriction. We have said that count nouns are quantized and mass nouns are divisive. These properties are not part of the lexical entries. The lexical entries are as before and we assume that we know for each noun meaning whether it is quantized or divisive.

Here is a meaning for the definite article that works for singular and plural count nouns and for mass nouns likewise.

(209) \( F(\text{the}_{(et)}) = \lambda w. \lambda P_{(et)} : \exists x[P(x) \& \forall y[P(y) \rightarrow y \leq x ] \& \forall x[P(x) \& \forall y[P(y) \rightarrow y \leq x ] \]

As an illustration we consider the LFs for the boy, the boys and the milk.

(210) a. \( [ [ \text{the boy} ] ] = \lambda w : \exists x[\text{boy}(w)(x) \& \forall y[\text{boy}(w)(y) \rightarrow y \leq x ] \& \forall x[\text{boy}(w)(x) \& \forall y[\text{boy}(w)(y) \rightarrow y \leq x ] \]
b. \( [ [ \text{the [PL boy]} ] ] = \lambda w : \exists x[\text{x is a boy-plurality in } w \& \forall y[\text{y is a boy-plurality in } w \rightarrow y \leq x ] \& \forall x[\text{x is a boy plurality in } w \& \forall y[\text{y is a boy-plurality in } w \rightarrow y \leq x ] \)
c. \[ \text{the milk} = \lambda w: \exists x \left[ \text{milk}(w)(x) \land \forall y \left[ \text{milk}(w)(y) \rightarrow y \leq x \right] \right] \]

Recall that the information between : and . is the presupposition. We can form pluralities of things in D_e by means of non-Boolean and:

(211) \( \text{Non-Boolean and} \)
\[ F(\text{and}_{\text{ee}}) = \lambda w. \lambda x. \lambda y. y + x \]

Here are some applications:

(212) a. \[ \text{Ann and [the PL boys]} = \lambda w. \text{the plurality that consists of Ann and the boys in } w \]

b. \[ \text{[the PL boys] [and the PL girls]} = \lambda w. \text{the plurality that consists of the boys in } w \text{ and the girls in } w \]

c. \[ \text{the milk and the coffee} = \lambda w. \text{the fusion of the milk in } w \text{ and the coffee in } w \]

In a mereology that knows of no absolute atoms we have a problem with counting. Consider the NP \text{six boys}. This should apply to a plurality \( x \) if \( x \) is the fusion of six boys. The individual \( x \) doesn’t give us the information that it is made up of boy atoms. To recover that information, we assume that numerals take a silent classifier as their first argument, where a classifier is simply a quantized predicate. The LF of \text{six boys} could then be:

(213) \[ \text{sixC [PL boy]} = \lambda w. \lambda x. x \text{ are boys in } w \land \left| \{ y \mid y \leq x \land C(y) \} \right| = 6, \]

where \( C = \text{person}(w) \)

The classifier is a free variable whose value is determined by the context. It should be an appropriate class of individuals, e.g. the persons in the world of evaluation. Here is the meaning of a numeral:

(214) A numeral
\[ \text{six} \text{ has the type } (et)(et, et). \]
\[ F(\text{six}) = \lambda w. \lambda C, C. \lambda P, C. x: C \text{ is quantized } \land x \text{ is a } C\text{-plurality. } P(x) \land \left| \{ y \mid y \leq x \land C(y) \} \right| = 6 \]

In classifier languages like Chinese or Japanese, a numeral is always accompanied by a classifier, e.g., Chin. \text{qi zhang zhuo zi} ‘six flat-thing table’, Jap. \text{ini san-biki} ‘dog three animal-thing’.

Let us take up plural predication next. Predicates can be classified according to the following criteria.
1. Purely distributive predicates: if a group has such a predicate, every member of the group has it. Examples: laugh, sneeze, frown.

2. Purely collective predicates: only groups can have them; a single individual cannot have such a predicate. Examples: gather, hate each other.

3. Mixed predicates: individuals can have them, but if the predicate applies to a group, its members cooperate in a particular way. Examples: eat the cake, lift the piano, write a paper.

Here are lexical entries for some of these predicates:

(215) A purely distributive predicate
F(laugh) = λw.λx: x is a person in w. x laughs in w.

Since “person” is a quantized predicate, laugh is only defined for singular persons. If we want to apply the verb to a plurality like the boys, we have to pluralize the predicate in a way that the plurality laughs if each member of it does. The plural operator * will do the job.

(216) A purely collective predicate
F(gather) = λw.λx: x is a plurality of animal-atoms in w. x gather.

“animal” is taken in the general sense of living being.

(217) A mixed predicate
F(eat the cake) = λw.λx. Each member of x eats a part of the cake and each part of the cake is eaten by a member of x. (The context has to make clear which are the relevant members of x.)

Given lexical entries of this sort we can analyse the following sentences:

(218) a. A boy laughed (singular subject, distributive predicate)
    b. The PL students gathered (plural subject, collective predicate)
    c. Ede/the PL boys ate the cake (singular or plural subject, mixed predicate)

None of the predicates contains a semantic plural. These examples make clear that the plural of the predicate is not licensed by a Plural operator modifying the predicate but by one contained in the subject. Hence we assume the following condition:

(219) The morphological plural of the verb is licensed by a PL-operator contained in the subject.
13.2. Plural of Verbs: Cumulation

But we cannot analyze the following sentences yet:

(220)   a. The boys laughed
         b. The boys and the girls met (separately)
         c. The boys ate a cake (each)

Each of these sentences has a distributive reading. It would seem then that the semantic plural of the VP is a distributive operator that distributes the predicates over the relevant parts of the subject, the singular boys in the case of sentence (a), the boys and the girls in the case of (b), and the particular boys in the case of (c). Such an approach is taken in (Link, 1991) and (Schwarzschild, 1996) (see 13.5). It works for the majority of cases but not for sentences like the following one:

(221)    The parents of Ann and Bill met.

This sentence requires a cumulation operator that applies to the relation noun parents. There is no way to obtain the correct interpretation by a distributor. Hence we will follow those who do the semantic plural of verbs by means of cumulation ((Krifka, 1989a), (Sternefeld, 1998), (Beck and Sauerland, 2000)).

(222)    The Cumulation Operator * (unrestricted version)

Let \( \varphi \) be any n-place predicate (of type \( e^n \)). Then \( *\varphi \) is of the same type.

\[
[[ *\varphi ]] \text{ is the smallest relation } R \text{ such that } [[ \varphi ]] \subseteq R \& \\
(\forall x_1, \ldots, x_n, y_1, \ldots, y_n)[R(x_1) \ldots (x_n) \& R(y_1) \ldots (y_n) \rightarrow R(x_1 + y_1) \ldots (x_n + y_n)]
\]

We will assume that * expresses the semantic plural for verbs (a standard assumption) and relational nouns. Note that * is a logical symbol. We could define functors, but we would have to do this for each type of relation separately. (Sternefeld, 1998) introduces the functor * for one-place relations, ** for 2-place relations, and so on. Our star symbol stands for any of these. The type of the modified expression makes it clear which star is used in each particular case.

With the exception of (220b) we can now analyse the recalcitrant plural statements just mentioned.

(223)    [the PL boys] *laughed   LF for (220a)

Suppose the boys are b1, b2 and b3, and the predicate laughed contains these. It follows
that \( b_1 + b_2 + b_3 \) is a member of the predicate *laughed*.

Sentence (220c) is analysed along the same lines. We assume that the boys are as before, the girls are \( g_1 + g_2 \), and \textit{met} denotes the set \{\( b_1 + b_2 + b_3 \), \( g_1 + g_2 \), \ldots\}. It follows that the LF is true in this scenario. The LF for (220c) is more interesting.

\[
(224) \quad [\text{the PL boys}] \quad *\lambda_1[a \text{ cake } \lambda_2[t_1 \text{ ate } t_2]] \quad \text{LF for (220c)}
\]

Suppose that \textit{ate} denotes the set \{\( <b_1, c_1> \), \( <b_2, c_2> \), \( <b_3, c_3> \), \ldots\}, \textit{cake} denotes \{\( c_1, c_2, c_3, \ldots \)\} and the boys are \( b_1 + b_2 + b_3 \). Then the predicate \( \lambda_1[a \text{ cake } \lambda_2[t_1 \text{ ate } t_2]] \) denotes the set \{\( b_1, b_2, b_3, \ldots \)\}. Hence the plural predicate *\( \lambda_1[a \text{ cake } \lambda_2[t_1 \text{ ate } t_2]] \) denotes the set \{\( b_1 + b_2 + b_3, \ldots \)\}. It follows that the LF is true in this scenario.

The LF shows that we pluralize the VP in the syntax: we first QR the object, then we QR the subject and insert the *-operator between the subject and the abstract created by the latter application of QR.

\[
(225) \quad [\text{the [ *[parents_{et}] (of) Ann and Bill] met}] \quad \text{LF for (221)}
\]

Suppose that \textit{parents} denotes \{\( <m_1, a>, <f_1, a>, <m_2, b>, <f_2, b>, \ldots \)\}, \textit{Ann and Bill} denotes \( a+b \), and \textit{met} denotes \{\( m_1+f_1+m_2+f_2, \ldots \)\}. The reader should verify for herself that this scenario makes the LF true.

The indefinite article of plural DPs is covert, and we will represent it by \( \exists \), which expresses the existential quantifier, which we have introduced in (115):

\[
(226) \quad \begin{align*}
\text{a.} & \quad \text{Caroline saw two toucans.} \\
\text{b.} & \quad [\exists \text{ twoC PL toucan}] \quad *\lambda_1 \text{ Caroline saw } t_1
\end{align*}
\]

Suppose \textit{toucan} denotes \{\( t_1, t_2, t_3 \)\} and \( \lambda_1 \text{ Caroline saw } t_1 \) denotes \{\( t_1, t_3 \)\}. In this scenario, the LF is true. Note that the morphology of the verb is singular, but the distributive reading requires a semantic plural. We have applied Pluralization in the syntax, but we would have obtained the same result if we had applied the *-operator to the verb \textit{saw} directly.

The following sentence has been brought up by (Scha, 1984). It shows that without the *-operator (or an appropriate distributor) it is impossible to obtain a compositional semantics that respects the syntax:

\[
(227) \quad \text{The sides of A are parallel to the sides of B.}
\]

\[
[\text{the sides of A} \quad [\text{the sides of B}] \quad *\text{parallel}
\]

Assume that A and B are rectangles. Suppose that \textit{sides of A} and \textit{sides of B} denote the sets
\{a_1, a_2, a_3, a_4\} and \{b_1, b_2, b_3, b_4\}, respectively; parallel is the set \{<a_1,b_1>, <a_1,b_3>, <a_3,b_1>, <a_3,b_3>, <a_2, b_2>, <a_2, b_4>, <a_4, b_2>, <a_4, b_4>, \ldots\}. Hence *parallel certainly contains the pair \langle a_1+a_2+a_3+a_4, b_1+b_2+b_3+b_4\rangle consisting of the sides of A and those of B.

If we allow to insert the *-operator wherever it makes sense, we can generate many LFs for a plural sentence, some of which are equivalent. Here is an illustration.

\[(228)\]

\begin{itemize}
  \item a. Ann and Bill own three computers.
  \item b. \[\exists \text{ threeC PL computers} \; \lambda_1 \; [\text{Ann and Bill}] \; \text{own} \; t_1\]
  \item c. \[\exists \text{ threeC PL computers} \; *\lambda_1 \; [\text{Ann and Bill}] \; \text{own} \; t_1\]
  \item d. \[\exists \text{ threeC PL computers} \; *\lambda_1 \; [\text{Ann and Bill}] \; *\lambda_2 \; t_2 \; \text{own} \; t_1\]
  \item e. \[\text{Ann and Bill} \; *\lambda_2 \; \exists \text{ threeC PL computers} \; *\lambda_1 \; t_2 \; \text{own} \; t_1\]
  \item f. \[\exists \text{ threeC PL computers} \; \lambda_1 \; \text{Ann and Bill} \; [*\text{own}] \; t_1\]
\end{itemize}

(b) represents a collective owning of three computers. (c) represents a distributive scenario that is equivalent with (b) on semantic grounds: if Anna and Bill collectively own three singular computers, then they own the group and vice versa. (d) describes a distributive scenario, which is equivalent to the former: both Ann and Bill own each a group of three computers. (e) is a distributive reading: each of Anna and Bill owns three computers, i.e., the two may own six computers. (f) is a particularly interesting LF, because it can be made true by rather different scenarios: (i) Ann and Bill commonly own the three computers; (ii) Ann and Bill own a computer each and a third one in common; (iii) Ann owns two computers and Bill owns one; (iv) Bill owns two computers and Ann owns one; (v) Ann owns two computers and Bill and Ann collectively own one; (vi) a scenario in which Bill owns one computer and Ann and Bill collectively own one.

The fact that the cumulation of the verb yields such a multiplicity of meanings might suggest that the *-operator only applies to the verb directly and the other stars are not permitted. Such an assumption follows without stipulation if we assume that verbs (and simple predicates in general) are only cumulated in the lexicon. Among others, (Krifka, 1986), (Winter, 2000) and (Kratzer, 2005) assume this. This amounts to the claim that there is no semantic singular-plural distinction for verbs. This, however, cannot account for all plural readings. For instance, the distributive reading represented in (228a) requires cumulation in the syntax (or something equivalent, i.e., an appropriate distributor).

Heim gives arguments in different papers, e.g. in (Heim, 1994c), that different readings of plural sentences are not simply due to lexical cumulation of a verb, which makes
the meaning vague. We have genuine ambiguity here. Here are some examples.

(229) A: The apples cost $2.
    B: No they didn’t, they cost $12.
    A: I meant $2 each.

A has a distributive reading in mind, B a collective one.

    B: So you spent $72 on apples. Are you out of mind?
    A: I meant all of them together.

A has a collective reading in mind, B a distributive one.

(231) A: The apples cost $2.
    B: That is not much for a whole dozen.
    A: I meant $2 per tray.

The differences in interpretation cannot be left to the facts plus the assumption that the predicate cost is lexically cumulated. We rather have to pluralize the complex VP appropriately. The unrestricted *-operator is not sufficient for that purpose. We have to relativize it to a contextually given cover.

(232) Cover (mereological version)

C is a cover of x – Cover(C,x) – iff C is a set such that \( \sum C = x \).

If C is a cover of the big thing U we say that C is a cover simpliciter – Cover(C). The relativized *-operator can now be defined as:

(233) The Cumulation Operator(s) \( *_C \) (restricted version)

Let \( \varphi \) be any n-place predicate (of type en) and let C be a cover variable. Then \( *_C \varphi \)

is of the same type.

\( [[ *_C \varphi ]] \) is the smallest relation R such that \( [[ \varphi ]] \subseteq R \& \\
( \forall x_1, \ldots, x_n, y_1, \ldots, y_n)[R(x_1) \ldots (x_n) \& R(y_1) \ldots (y_n) \& C(x_i) \& C(y_i) \rightarrow \\
R(x_1+y_1) \ldots (x_n+y_n)], i = 1, \ldots, n \)

The three readings in the examples discussed can now be analysed as follows:

(234) a. [the PL apples] \( *_C \) cost $2

C = \{ x \mid x \text{ is an apple in } w \}

b. [the PL apples] \( *_C \) cost $6
C = \{ x \mid x = \text{the apples in } w \} 

c. [the PL apples] \*C cost $2

C = \{ x \mid \exists y [ y = \text{a tray in } w \& x = \text{the content of } y \text{ in } w ] \} 

Whether a plural statement is true or not thus depends on the context that determines the value of the cover variable. For the examples given it suffices to assume that C is a cover of the subject. The examples (a) and (b) have the most fine-grained and the most coarse cover that give us the set of apples and the entire group respectively. These are the unmarked interpretations of C. The cover in (c) is a marked option.

(Schwarzschild, 1996) suggests generalizing covers to pair covers:

(235) \textit{Pair Cover}

C^2 is a pair cover of \langle x, y \rangle \text{ – PairCover (C,x,y) – iff } x = \Sigma \{ x' \mid (\exists y') \text{ C}(x')(y') \} \& y = \Sigma \{ y' \mid (\exists x') \text{ C}(x')(y') \} 

The definition can be generalized to covers of any number of places in an obvious way. We will write n-place covers as C^n. The definition of restricted cumulation has to be revised accordingly: the conditions C(x_i) and C(y_i) have to be replaced by C^n(x_1)…(x_n) and C^n(y_1)…(y_n), respectively.

The revision is necessary for describing scenarios like the following:

(236) The sides of the bookcase are parallel to the walls. (Heim)

[the sides of the bookcase] are [*C^2 parallel] [to the walls]

C^2 = \{ \langle b1, w1 \rangle, \langle b2, w2 \rangle, \langle b3, w3 \rangle \} 

The LF is true in the first scenario but not in the second:

This is so because the cover pairs each side of the bookcase with the wall fronting it. It is obvious that e.g. \langle b1, w1 \rangle is a member of \[ C^2 \parallel \] , but \langle b3, w1 \rangle isn’t. For the scenario B, this is required, however.

Pair covers are also useful for generating co-distributive readings:

(237) Ann and Bill [*C^2 wrote] two papers

The scenario we have in mind is that Anna wrote paper 1 and Bill wrote paper 2. For the LF
we use the pair cover $C^2 = \{<a, p1>, <b, p2>\}$, which blocks readings that are not co-distributive. One might dispute that this is a reading and $C^2$ is therefore perhaps not motivated by this datum, but we certainly need $C^2$ for cases like (236), which cannot be analysed by 1-place covers.

(Beck and Sauerland, 2000) give examples that cannot be analysed by assuming lexical cumulation alone:

(238) The boys gave the girls a flower

\[
[\text{the boys}] [\text{the girls}] *_{\lambda_2 \lambda_1} [a \text{ flower } \lambda_3 [t_1 \text{ gave } t_2 t_3]]
\]

The scenario is this: the boys are b1 and b2, the girls are g1 and g2 and the flowers are f1 and f2 and the extension of gave is \{<b1, f1, g1>, <b2, f2, g2>\}. The LF is generated by applying three times QR and inserting the *-operator under the indirect object. The indirect object is QR-ed under the subject. Another example is:

(239) The two men want to marry the two women.

\[
[\text{the two men}] [\text{the two women}] *_{\lambda_2 \lambda_1} t_1 \text{ want } [\text{PRO to marry } t_2]
\]

The scenario is: John wants to marry Alice, and Bill wants to marry Janice.

Certain examples brought up by (Schein, 1993) are not analysed by cumulation but by an appropriate decomposition of the verb, here find as CAUSE to be found.

(240) The copy editors found every mistake in the manuscript.

\[
[\text{the copy editors}] \text{ CAUSED every mistake in the manuscript (is) found}
\]

Let us finally comment on example (204) here repeated as:

(241) The boys scratched their noses.

\[
[\text{the PL boys}] *_{\lambda_1} t_1 \text{ scratched } [\text{their noses}]
\]

The object contains no semantic plural at all. No boy has more than one nose. The plural morphology is licensed by the semantic plural of the subject under binding. This sort of dependent plural is possible for variables semantically bound by a plural antecedent. For details, see (Heim, 2005) and (Kratzer, 1998).

13.3. Reciprocity

The semantics of the reciprocal pronoun is still under debate. We present a simplified
version of the account in (Heim et al., 1991).

(242) The reciprocal pronoun each other

\[ \lambda w \lambda x \lambda y . tz [z \neq x \& z \leq y] \]

“those z of y that are different from x”

The reciprocal pronoun depends on two arguments, where the first argument x is the contrast argument, and the second argument y is the range argument. The arguments are represented as variables. Chomsky’s Binding Theory states that the binding of reciprocals obeys Principle A (see section 15). So the most natural assumption is that both variables are locally bound by the same antecedent (cf. HLM). This requires a twofold application of QR to the antecedent.

(243) John and Bill and Mary like each other.

\[ [\text{John} + \text{Bill} + \text{Mary}] \lambda z [t_2 *[\lambda_1 \text{ea}(1)(2) *[\lambda_2 t_1 \text{like } t_2]]] \]

Roughly: (\( \forall x \in J+B+M \)) x likes those z \( \leq J+B+M \) that are different from x.

The construal of the LF works as follows. First we QR the subject to the position indicated by \( t_2 \) thereby binding the contrast variable 1 of the reciprocal. (In Chomsky’s Binding Theory 1 would be an anaphor.) Then we QR the reciprocal under the subject. In the next step we “star” the abstracts created by these movements. The lower star is necessary because the verb like contains only pairs of singular individuals, but the subject has to like “the others”, a plurality. The higher star is necessary in order to obtain the distribution of the reciprocal predicate over the subject. The subject of this predicate is always a singular individual. In the final step we QR the subject in order to bind the range variable 2 of each other. This variable typically ranges over pluralities.

The LF describes a rather strong kind of reciprocity: John likes Bill and Mary, Bill likes John and Mary, and Mary likes Bill and John. The reader should convince himself that we have to QR the reciprocal and star the abstract it creates.

While local binding for the contrast variable seems uncontroversial, there is evidence that the range variable occasionally requires a more remote binding:

(244) John and Mary say they will defeat each other. (Higginbotham)

‘John says: “I will defeat Mary”, and Mary says: “I will defeat John.”’

\[ [J + M] \lambda z t_2 *[\lambda_1 t_1 \text{say } t_1 \text{defeat } \text{ea}(1)(2)] \]

The antecedent of the range variable may even be contained in an island:
(245) The people who worked for Street and Weinberg thought they would defeat each other. (Dimitriadis, 2000)

‘The people who worked for Street think: “We will defeat the people who worked for Weinberg”, and the people who worked for Weinberg thought: “We will defeat the people who worked for Street.”’

\[ S + W \] \( \lambda_2 [\text{the people who worked for } t_2] \lambda_1 [t_1 \text{ thought they } t_1 \text{ would defeat } t_2] \]

Sometimes we do not find an antecedent for the range variable at all:

(246) John and Mary said their candidates would defeat each other. (Dimitriadis, 2000)

‘John said: “My candidate will defeat Mary’s candidate”, and Mary said: “My candidate will defeat John’s candidate.”’

\[ J + M \] \( \lambda_1 t_1 \text{ said } [\text{their } t_1 \text{ candidates }] \lambda_2 t_2 \text{ would defeat } t_2 \]

\( g(t_3) = \text{John’s candidate } + \text{Mary’s candidate} \)

Intuitively, the antecedent for the range variable 3 is determined by “their candidates”. The problem, however, is that this term is a semantic singular and can serve as the antecedent of the contrast variable 2 only. Examples like these suggest that the range variable should remain free and be determined by the context. This strategy is advocated in (Beck, 2001). We would have to reanalyse example (245) accordingly. It is easy to show that such an approach heavily overgenerates. So this theory is not the last word on the reciprocal.

13.4. Summary

The semantic plural operators are cumulation operators, one-place for non-relational count nouns and n-place for verbs. The operator **PL** is defined in a more complicated way than the *-operator, which is due to the fact that we need the notion of P-atom in order to say what a plurality is. DPs with plural morphology always contain a plural operator. The plural morphology of the finite verb is licensed by a subject under agreement. If a sentence has a plural object, the VP must be “starred” but doesn’t exhibit morphological plural.

- *-insertion is a rule of construal.

It applies in the syntax after the application of QR whenever the interpretation requires so. In order to generate the appropriate relations from the VP, QR has to apply in a parallel way, i.e., the object has to be QR-ed under the subject. The *-operator is usually restricted by a cover variable which might be an n-place cover. The reciprocal depends on two variables. The binding of the contrast variable obeys Principle A. The binding-theoretical status of the range
variable remains unclear.

13.5. Notes on the literature

The best introduction into plural semantics is the unpublished paper (Heim, 1994c). A theoretical foundation of mereology and its relation to set theory is (Lewis, 1991). The first detailed formalization of plural operators and their application to the analysis is found in the work of G. Link, e.g. (Link, 1991). Link introduces a one-place cumulation operator (* and \(\otimes\)). He discusses the examples due to (Scha, 1984), but he cannot analyze them, which is due to the fact that his theory contains no n-place cumulation. Link’s ontology assumes atoms, which makes the treatment of count nouns easy but has to assume a multiplicity of part-relations. Link makes use of cumulation in the syntax. (Krifka, 1989a, Krifka, 1989b) gives a unified treatment of mass and count nouns. The criteria for countability are provided by measure functions, e.g. 6 liters applies to an instance of milk if its volume is 6 liters. Krifka introduces n-place cumulation but only at the lexical level. The distinction between count and mass nouns in terms of divisibility is due to (Quine, 1960). (Scha, 1984) has brought the attention to plural sentences with more than one quantifier that have cumulative readings. His solution does not assume a general cumulation operator. He encodes the required quantifiers into the lexical entry. For instance, the adjective parallel would have the semantics \(\lambda x.\lambda y.(\forall x' \leq x)(\exists y' \leq y)[x' \text{ is parallel to } y'] \& (\forall y' \leq y)(\exists x' \leq x)[x' \text{ is parallel to } y']\). Scha doesn’t have a theory that derives the possible quantifier combinations for relational lexical entries. The most extensive analysis of cumulative readings and related phenomena is found in (Schwarzschild, 1996). His account is somewhat different from the analysis given here. He lexically cumulates every verb in the plural. Instead of relativized cumulation operators, he works with relativized distributors. For instance, a two-place relativized distributor is given by the following definition:

\[
(247) \quad \text{A two-place distributor} \\
\text{DIST}^2 = \lambda w.\lambda X_1.\lambda X_2.\lambda x.\lambda y. \text{PairCover}(x,y) \cdot (\forall x')((\forall y') \cdot \text{Cover}(x',y')) \cdot (\exists x')((\exists y') \cdot \text{Cover}(x',y')) \cdot (\forall y') \cdot (\forall x') \cdot (\forall y') \cdot \text{Cover}(x',y') \Rightarrow \text{Cover}(x',y')
\]

The co-distributive reading of the sentence in (237) would be represented as:

\[
(248) \quad [\text{Ann and Bill}] [\text{et}\exists \text{ two}\text{C}1 \text{ PL papers}] [\text{et}\text{C}2 \text{ DIST}^2 \text{C}2 \text{ et}\text{C}1 \text{ t}1 \text{ wrote t}2]]
\]

\[
\text{C}1 = \{x \mid x \text{ is a paper in } w\} \\
\text{C}2 = \{<\text{Ann}, \text{paper}1>, <\text{Bill}, \text{paper}2>\}
\]
Lexical cumulation comes for free and the distributors play the role of our relativized $^*_C$-operator. In most cases, lexical cumulation is not needed, but examples like (221) require it. Schwarzschild’s covers always cover the entire universe, which is not implemented in the rules given here. Schwarzschild’s approach and the one given in this article are largely equivalent. The generalized semantics for the definite article in (210) traces back to (Sharvy, 1980). The technique of restricting star operators or distributors by means of covers is presumably due to (Heim, 1994c). The analysis of reciprocal outlined here is not precisely that in (Heim et al., 1991), it is close to (Beck, 2001). Recent proposals to overcome the difficulties discovered by (Dimitriadis, 2000) are found in lecture notes by I. Heim. (Sternefeld, 1998) doesn’t treat the reciprocal as a nominal but as an operation that applies to a relation. For instance, the VP like each other would have the representation $\lambda z[\forall x \forall y (x \neq y \& \text{like}(y)(x))](z)(z)$. The section has not treated pluriactional adverbs like in turn in “The girls served the guests in turn”. These require the introduction of events and a refinement of the plural operator; see (Lasersohn, 1995) and (von Stechow and Beck, 2007), among others.

14. NEGATION AND NEGATIVE INDEFINITES

14.1. Sentence negation

A classic in the syntax of negation is (Jespersen, 1917). Jespersen thinks that sentential negation should occur in a preverbal position because it is the verb or the entire sentence that is negated. At a stage, Jespersen is wondering why German has negation at the end of a sentence in many cases. He thinks that this is not logical. As a simple example consider the following sentence:

(249) Das glaube ich nicht.

that believe I not

It looks as if the negation occurs at a sentence final position. A closer inspection, however, shows that this is not so. Every syntactician of German would agree that the analysis of the sentence is something like this:

(250) DS: [CP C [VP nicht [VP ich das glaube]]]

SS: [CP das$_1$ [C: glaube$$_2$$ [VP ich$_3$ [VP nicht [VP t$_3$ t$_1$ t$_2$]]]]]

The object is moved to [Spec, CP]. The finite verb is moved to C. The subject is moved to the subject position (or simply scrambled out of the VP; referential terms never occur in the scope of the negation nicht in German at s-structure). The negation particle nicht is not
moved. If we disregard tense, it occupies a sentence initial position, exactly as Jespersen wants to have it.

All these movements are reconstructed at LF and we are left with the configuration given by the DS.

Negation itself has the semantics we know from propositional logic:

\begin{equation}
\text{Negation}
\end{equation}

Germ. \textit{nicht}, Engl. \textit{not}, French \textit{pas},... all mean $\lambda w.\lambda T_T.0$, if $T = 1$; 1, if $T = 0$.

So the semantics of negation is very simple, but the syntax interacting with the negation word might be complicated and obscure the view. Negative particles like \textit{nicht} are not the only way to express negation. Negative indefinites enter the scene, and here the picture gets complicated.

14.2. Negative Indefinites

One of the classical achievements of the theory of Generalised Quantifiers is the analysis of negative indefinites as negative quantifiers:

\begin{equation}
\text{A negative indefinite as a negative quantifier (Montague, 1970c)}
\end{equation}

$F(\text{\textit{nobody}}_{et}) = \lambda w.\lambda Q_{et}. \neg (\exists x)[\text{Person}(x,w) \land Q(x)]$

There is recent linguistic evidence that this analysis is not tenable for natural languages. This newer view suggests that \textit{nobody} and related words rather are existential quantifiers carrying the feature [u-NEG], which has to be licensed by a semantic negation carrying the feature [i-NEG].

\begin{equation}
\text{A negative indefinite as an n-word}
\end{equation}

$F(\text{\textit{nobody}}_{et}) = \lambda w.\lambda Q_{et}. (\exists x)[\text{Person}(x,w) \land Q(x)]$

The word carries the feature [uNEG].

Languages differ in the following way: (a) The semantic negation can license one/many n-words; (b) the licenser is overt/covert; (c) Multiple Agree is obligatory/optional. Multiple Agree means that one feature [iNEG] may license more than one feature [uNEG].

Negative indefinites (NIs) can belong to different logical types. Here is a list of some possibilities:

\begin{equation}
\text{Some NIs}
\end{equation}

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>German</th>
<th>Norwegian</th>
<th>Italian</th>
<th>French</th>
<th>Russian</th>
</tr>
</thead>
</table>

90
The classical analysis would analyse all these as negative universal quantifiers. For instance, never would have the following meaning: \( \lambda w. \lambda t. \lambda P_t. \neg (\exists t') [t' \subseteq t \land P(t')] \). The n-word account has this meaning without negation of the existential quantifier. In the following we will not speak about temporal quantifiers.

### 14.3. Negative Concord (NC) vs. Double Negation (DN)

The theory of NIs is mainly concerned with two phenomena, Negative Concord (NC) and Double Negation (DC), which are illustrated by the following examples:

(255) Negative Concord (NC)

a. Masha nikogo ne videla. (Russian)
   Masha n-person not saw
   ‘Masha didn’t see anyone.’

b. Nikto nichego ne videla.
   n-person n-thing not saw
   ‘Nobody saw anything.’

NC is the phenomenon that we find several morphological negations but only one semantic negation.

(256) a. Maria non ha visto nessuno. (Italian)
   Maria not has seen n-person
   ‘Mary didn’t see anybody.’

b. Nessuno ha visto niente.
   n-person has seen n-thing
   ‘Nobody saw anything.’

Russian (together with other Slavic languages and Modern Greek) is a strict NC-language, whereas Italian (Spanish, Portuguese) is a non-strict NC-language. The former has a negated verb, whereas the latter has a negated verb only if there is no n-word in preverbal
position.

In Double Negation (DN) languages, each NI contributes a semantic negation:

(257) Nobody didn’t come.
     = Everybody came

(258) Dieses Jahr hat kein Student nicht bestanden.  (German: (Penka, 2007))
     ‘This year, no student didn’t pass.’

14.4. Analysis of Negative Concord

For the analysis of NC-languages, we stick to the theory of (Zeijlstra, 2004), which assumes an abstract semantic negation $Op_{-}$ carrying the feature [iNEG]. It checks all the features [uNEG] under c-command and within a certain local domain (no finite clause may intervene). Here is the analysis of one of the Russian examples.

    LF: $\neg$someone $\lambda_1$ something $\lambda_2$ t$_1$ saw t$_2$

The LF shows the essential outcome: we have three morphological negations in the syntax but only one semantic negation. Russian belongs to the strict NC languages, which have the following characteristics: (i) The semantic negation $Op_{-}$ is covert; (ii) the negation marker at the verb is semantically void; (iii) there is multiple agreement.

In contradistinction, in a non-strict NC-language such as Italian, the verbal negation is semantically interpreted and licenses post verbal NIs via multiple agreement. NIs that occupy a position in front of the negated verb are somewhat degraded with respect to acceptability. They contribute to a DN-reading. Sentences without negation markers may contain a preverbal and post verbal NI and have an NC-reading. NIs in preverbal position are licensed by a covert semantic negation.

Here are the relevant configurations:

(260) A non-strict NC-language (Italian)
       a. Maria non[iNEG] ha detto niente[uNEG] a nessuno[uNEG].
           $\neg$something $\lambda_1$ somebody $\lambda_2$ Mary said t$_1$ to t$_2$
       b. $^2Op_{-}$[iNEG] Nessuno[uNEG] non[iNEG] ha detto niente[uNEG].
           $\neg$somebody $\lambda_1$ $\neg$something $\lambda_2$ t$_1$ said t$_2$
           $\neg$somebody $\lambda_1$ something $\lambda_2$ t$_1$ said t$_2$
French negation is particularly challenging because the language has NC-patterns and DN-patterns.

(261)   Je n’ai pas faim.
        I NEG have NEG faim
        ‘I am not hungry.’

Of the two negations, ne is void and pas is interpreted. The following sentences contain the NI personne:

(262)   a. Jean n’a vu personne.
        Jean NEG has seen n-person
        ‘Jean didn’t see anyone.’

   b. Jean n’a pas vu personne.
        Jean NEG has NEG seen n-person
        ‘Jean didn’t see nobody.’

(262a) has an NC-reading, and (262b) has a DN-reading. The following sentence is ambiguous between a NC-reading and a DN-reading:

(263)   Personne n’aime personne.
        n-person NEG loves n-person
        a. ‘Nobody loves anybody.’

        b. ‘Nobody loves nobody’ = ‘Everybody loves somebody’

(Penka, 2007) presents a theory to derive the distribution. (i) The NI personne has to be licensed by a covert negation; it has the feature [uNEG-], and Op. has the feature [iNEG-]; NIs that can be licensed by either a covert or overt negation carry the feature [uNEG]. (ii) The negation marker ne has the feature [uNEG], whereas pas carries [iNEG]. (iii) Multiple Agree is optional.

We are now in a position to account for the French examples.

(264)   a.  \[NegP pas_iNEG] [Neg ne_iNEG] \[VP t_j [je ai faim]]

   b.  \[TP jelk [t [Neg ne_iNEG] aii]m] [NegP pas_iNEG] \[tm [VP t_j [tk ti faim]]]] \] (SS)

   c. LF: \(\neg\) I have hunger

The surface syntax is in the style of (Pollock, 1989): ne is the head of NegP, pas moves to [Spec, NegP], the finite verb moves to the head of TP through the head of NegP, and the subject moves to [Spec, TP]. At LF, pas-movement and all the head movements are
reconstructed and ne is deleted by FI. The configuration where pas can check [uNEG] of ne
is depicted in (264a).

(265) OP_¬[uNEG-] Jean n’[uNEG] a vu personne[uNEG-]

¬somebody λ1 Jean saw t1

The DN-reading of (262b) is due to the stipulation, that personne requires a covert
semantic negation as licenser. Therefore, the overt negation pas cannot license this n-word
and we have to assume an additional operator Op, in the scope of pas. The relevant
licensing configuration is this:

(266) Jean n’a pas vu personne.

pas[°[uNEG]] n’[uNEG] tj Op_¬[uNEG-] Jean a vu personne[uNEG-]

¬¬somebody λ1 Jean saw t1

The ambiguity of the sentence in (263) is due to the optionality of Multiple Agree in
French:

(267) Personne n’aime personne.

a. ¬Op[°[uNEG-]] personne[°[uNEG-]] n’[uNEG] aime personne[°[uNEG-]] (Multiple Agree)

¬somebody λ1 somebody λ2 t1 loves t2

b. ¬Op[°[uNEG-]] personne[°[uNEG-]] n’[uNEG] ¬Op[°[uNEG-]] tj aime personne[°[uNEG-]] (Local Agree)

¬somebody λ1 ¬somebody λ2 t1 loves t2

14.5. Analysis of DN

Let us turn to DN-languages next. The straightforward account seems to be that the negative
marker of the verb (= NM) and the NIs express semantic negations. Such a line is taken in
(Zeijlstra, 2004). There are, however, facts that speak against this theory, viz. the
phenomenon of split scope of NIs into a negation + indefinite.

(268) Bei der Prüfung muss kein Professor anwesend sein. (German)

at the exam must n-DET professor present be

a. ‘It is not required that there be a professor present’  ¬> must > ∃

b. ‘There is no professor who is required to be present.’ ¬∃ > must

c. ?? ‘It is required that there be no professor present’  must > ¬∃

The split reading cannot be reduced to wide scope of a negative quantifier. Sentences with
expletive *es*, a modal verb and an NI-subject only have the split reading:

(269) Es muss kein Arzt anwesend sein.

\[ \text{it must n-DET physician present be} \quad \neg \text{ > must > } \exists \]

‘It is not required that a physician be present.’

The NI can be part of an idiom:

(270) Du darfst ihm kein Haar krümmen

\[ \text{you may him n-DET hair bend} \]

‘You may not do him any harm whatsoever.’

It is obvious that examples of this kind cannot be analysed under the assumption that *kein Haar* is a negative quantifier because the term has no independent, literal meaning within an idiom.

Predicative nominals as objects of copulative verbs have the split reading if they are NIs:

(271) Ede wurde kein Ingenieur.

\[ \text{Ede became no engineer} \]

‘Ede didn’t become an engineer.’

Under topic-focus accent, a universal quantifier can take scope between the negative and the indefinite part:

(272) /JEder Student hat KEIN\ Auto.

\[ \text{every student has n-DET car} \]

‘Not every student has a car.’

The following analysis of German NIs is due to (Penka, 2007): (i) German NIs have the feature [uNEG-]. (ii) OP\_\{NEG\_\} is an adverb and can be inserted in front of any expression of type t. (iii) German doesn’t have Multiple Agreement. (iv) OP\_ and NI must be adjacent at SS.

In what follows, we give an analysis of the German examples.

(273) OP\_\{NEG\_\} [kein\{uNEG\_\} Student\_1 hat\_2 nicht\{NEG\_\} t\_1 bestanden t\_2.

\[ \neg \text{some student } \lambda_1 \neg t_1 \text{ passed} \]

Note that the features are licensed at SS. Even if we reconstructed the NI *kein Student* to its base position, it couldn’t be licensed there because NIs require an abstract licenser, in
other words, the negation adverb nicht can never license an n-word. Thus we obtain a DN-reading. Similarly, we have a DN-reading in the following example:

(274) weil OP_{NEG-} [kein_{NEG-} Gast]_1 OP_{NEG-} [kein_{NEG-} Geschenk]_2 [t_1 t_2 mitbrachte]
because n-DET guest n-DET present brought

¬some guest λ₁ ¬some present λ₂ [t₁ brought t₂]

The following examples exhibit the split scope phenomenon, i.e., ¬ is above a modal and ∃ is below it.

(275) OP_{NEG-}[[kein_{NEG-} Professor anwesend sein] muss]
¬must[some professor be-present]

(276) Ede₁ OP_{NEG-}[t₁ [kein_{NEG-} Ingenieur] wurde]
¬Ede BECOME an engineer

The meaning of BECOME could be λ.w.λQ_{beg(t)}.λ.x.λt.¬Q(λ.y.y=x)(beg(t)) & Q(λ.y.y=x)(end(t)).

The following example is particularly fascinating because it shows the interaction of NI-licensing and reconstruction at LF:

(277) /JEder Student₁ hat₂ OP_{NEG-} t₁ KEIN\ Auto t₂
¬every student λ₁ a car λ₂ t₁ has t₂

To obtain this LF, we have to reconstruct the topicalized subject to its trace – with subsequent QR of the subject and the object.

Penka’s theory is corroborated by Scandinavian data:

(278) a. Jon leser ingen romaner. (Norwegian)
Jon reads n-DET novels
‘Jon doesn’t read (any) novels.’
b. *Dette er en student som leser ingen romaner
this is a student that reads n-DET romaner
c. *Jon har lest ingen romaner
Jon has read n-DET novels

In Skandinavian, NIs are only grammatical if they are adjacent to a position the negative marker ikke can occupy. (279a) is the only example meeting this condition:

(279) a. Jon leser ikke noen romaner.
Jon reads NEG some novels
b. Dette er en student som ikke leser noen romaner.
   this is a student that reads NEG some novels
c. Jon har ikke lest noen romaner.
   Jon has not read some novels

The data follow from Penka’s theory under the assumption that Scandinavian NIs are analyzed as in German. Furthermore, we make the standard assumption that the finite verb moves to C in main clauses but not in subordinate clauses. Here is the analysis of the examples in (278):

(280)  a. \[ CP \text{Jon} \text{1 leser} \text{2 OP}_{\neg \text{[uNEG-]}} [VP t_1 t_2 \text{ingen}_{\text{uNEG-}} \text{romaner}] \].  (Norwegian)
   ‘Jon doesn’t read (any) novels.’
b. *Dette er en student \[ CP \text{CP} \text{1 som OP}_{\neg \text{[uNEG-]}} [IPI t_1 \text{leser ingen}_{\text{uNEG-}} \text{romaner}] \]
   this is a student that reads n-DET romaner
c. *\[ CP \text{Jon} \text{1 har2 [VP t_2 [OP}_{\neg \text{[uNEG-]}} [\text{PartP} t_1 \text{lest ingen}_{\text{uNEG-}} \text{romaner}]]] \]
   Jon has read n-DET novels

The analysis ignores the IP (and possibly other projections) between CP and VP. We see that verb movement created adjacency for example (280a) but not for the other constructions.

14.6. Summary

Negative indefinites always seem to be existential quantifiers. Strict NC languages differ from non-strict NC in the nature of the negative marker of the verb: in the former the marker is semantically empty and behaves like a NI, in the latter it is a semantic negation and licenses a NI. DN languages require an abstract semantic negation for each NI. NI might differ in requiring an overt or a covert licenser. The analysis of NC languages requires multiple agree. The construal of negated sentences is rather simple. Negative indefinites are generalized quantifiers that have to be QR-ed for the familiar reasons.

14.7. Notes on the literature

The account outlined in this section is due to (Zeijlstra, 2004) and (Penka, 2007). I refer the reader to these references for a discussion of different approaches to negation. Here, I will
mention only a few papers that are related to this chapter. In generative syntax there seems to be three leading ideas for the analysis of NIs. The first is that NIs are negative quantifiers. The second is that they are moved at LF to a certain operator position, say the Negation Phrase (NegP). This movement is thought in analogy to WH-movement at LF. The third idea is that NIs are Negative Polarity Items (NPIs). Different approaches combine these ideas in a different way. None of these approaches is compatible with the theory outlined here, though the third approach comes closely to our view. So let us start with that.

We haven’t treated NPIs in this article. Examples of NPIs are any, anybody, anything, anywhere. The most influential theory accounting for the distribution of these is (Ladusaw, 1979). The theory says that an NPI is licensed when it occurs in a downward entailing (DE) context. A function f is downward entailing if f(A) entails f(B) iff B \subseteq A.

(Laka, 1990) considers NIs like Spanish nadie ("nobody") as an NPI that has to be checked in an operator position \( \Sigma \) that contains a negation. If \([\text{Spec}, \Sigma]\) is filled by a NI, the negation has to be abstract. Thus we have the contrast (the notation Op \( \neg \) is not used by Laka):

\[
\begin{align*}
\text{a. } & \left[ x_2 \text{ nadie}_1 \text{ Op}\_ \left[ t_1 \text{ vino}\right]\right] \\
& \text{nobody came} \\
\text{b. } & \left[ x_2 \text{ no}_\neg \left[ \left[ t_1 \text{ vino} \right] \text{ nadie}_1\right]\right] \\
\end{align*}
\]

If the negation marker could somehow extend its scope over the NI in the specifier and NIs were interpreted as existential quantifiers, then this would be a viable analysis. (Laka doesn’t commit herself to the details of interpretation.) One of the problems arising with this analysis is the extension to strict NC languages: there the negative marker of the verb can’t express a semantic negation. Another problem is the different distribution of NIs and NPIs. A detailed discussion pointing out more problems is found in (Penka, 2007).

A standard reference for the syntax/semantics of negation in Italian is (Zanuttini, 1991) and (Haegeman and Zanuttini, 1991). According to this theory, NIs are universal negative quantifiers that have to be moved to \([\text{Spec}, \text{Neg}]\) to check a particular negation feature carried by them. In this system nessuno ("nobody") means \( \lambda w. \lambda P. x. (\forall x)[\text{person}(w,x) \rightarrow \neg P(x)] \). The sentence

\[
\begin{align*}
\text{Nessuno ha parlato con nessuno.} \quad \text{(Italian)} \\
& \text{nobody has talked with nobody} \\
& \text{“Nobody talked to anybody”}
\end{align*}
\]

has the LF
The abstract negation is the head of NegP and checks the features of the two NIs. If we evaluate the this LF by means of FA, we obtain the wrong reading, namely:

\[ \lambda w.(\forall x)[\text{person}(w,x) \rightarrow \neg (\forall y)[\text{person}(y,w) \rightarrow \neg \neg x \text{ talked to } y \text{ in } w]] \]

= “Everybody didn’t talk to everybody.”

In order to avoid this, Zanuttini introduces an absorption rule that deletes all the negations in front of the quantifiers leaving only the sentential negation (cf. (Haegeman and Zanuttini, 1991: p. 139). I don’t know of any method in semantics to make such a rule precise. In any case, the approach would be entirely non-compositional.

(Giannakidou, 1997) classifies NIs as NPIs as well. She says that NIs are licensed by an anti-veridical operator. Op is anti-veridical if Op(p) entails \( \neg p \). Unlike Ladusaw, Giannakidou defines NIs as universal quantifiers, e.g. \( F(\text{KANENAS}) \) (Greek “nobody”) = \( \lambda w.\lambda P.(\forall x)[\text{person}(x,w) \rightarrow P(x)] \). It is stipulated that NIs out-scope their licenser at LF. The movement is triggered by the licenser dhen (“not”) and is thus an analogon to WH-movement. Here is an example:

\[ \text{KANENAS dhen ipe TIPOTA} \quad \text{(modern Greek)} \]

nobody not said nothing

\[ \text{SS: } [\text{OpP nobody}_1 \text{ not } [\text{VP t}_1 \text{ said nothing}]] \]

\[ \text{LF: nothing}_2 [\text{OpP nobody}_1 \text{ not } [\text{VP t}_1 \text{ said t}_2]] \]

\( \lambda w.\forall y[\text{thing}_w(y) \rightarrow (\forall x)[\text{person}_w(x) \rightarrow \neg \text{said}(x,y)] \]

Note that the negation marker has semantic content in this approach. Thus Giannakidou’s LFs look similar to those of Zanuttini, but they can do the semantics compositionally. No absorption rule is needed. One of the problems with this approach is that it cannot treat non-strict NC languages. Recall that the corresponding Italian sentence \textit{Nessuno non ha detto niente} has a DN reading. Another problem is that the theory cannot generate \textit{de dicto} readings in NC languages. Penka mentions the Ukrainian sentence:

\[ \text{Poranenomu ne potriben nijakyj likar.} \quad \text{(Ukrainian)} \]

injured-dat not required no doctor

“The injured doesn’t need any doctor”

The analysis of Giannakidou requires talk about particular doctors.

(Herburger, 2001) investigated non-strict NC-languages, especially Spanish. She holds
the view that NIs are ambiguous: if an NI occurs in front of a negated verb, it is a negative universal quantifier. If it occurs in the scope of the verbal negation, it is something like an existential quantifier. One of the problems of such an approach is to determine the distribution of NIs. For a detailed criticism, see (Penka, 2007: 2.2).

15. **Getting rid of QR?**

In the theory of LF outlined in this article, QR plays a central role. In most cases, the rule is covert in the sense that it applies at the LF branch. As it stands, the rule is unconstrained and overgenerates in the sense that it produces LFs that have unattested readings. Much of the work done in Generative Grammar is devoted to constraints on overt movement, i.e. movement visible at SS. Let us therefore ask whether we need QR at all.

QR was motivated by three tasks that had to be performed: (i) Quantifiers in object positions had to be QR-ed for type reasons; (ii) scope ambiguities must be resolved by QR; (iii) variable binding, i.e., the binding of pronouns requires QR.

The three phenomena are interrelated. In many cases the application of QR is so local that we can eliminate the rule from the grammar and replace it by lexical rules or special principles of composition. In such cases the problem of constraining QR vanishes because there is no QR in the grammar anymore. But this is not always possible. Scope ambiguities cannot always be treated lexically. This speaks against the elimination of QR from the grammar. We keep the rule and consequently have to speak about restrictions.

15.1. **Quantifiers as objects: Type lifting?**

Let us talk about the problem of the object first. We assumed that a quantifier in object position had to be QR-ed for type reasons. A method to obtain the same result consists in type-lifted entries for verbs:

(288)  
\[ F(\text{knows}_\text{et}(\forall \text{t}))(\text{et})) = \lambda w.\lambda Q_{\text{et}}.\lambda x.\lambda y. x \text{ knows } y \text{ in } w \]

If we assume this entry we can interpret the SS of the sentence without QR-ing the object.

(289)  
\[ [\text{Barbara}_{\text{et}} \text{ knows } [\text{every linguist}]] \]
\[ = \lambda w. (\forall x) [\text{If } x \text{ a is a linguist in } w, \text{ then Barbara knows } x \text{ in } w] \]

Thus the problem of the object doesn’t require QR. There is, however, a price to be paid.
The lexicon entries are difficult to understand and they obscure the fact that intuitively a verb like *knows* simply expresses a two place-relation between individuals. Furthermore the elimination of QR is only spurious. As the inspection of the lexical entry of the verb shows QR is put into the lexical entry. An advantage of this method is that it constrains QR to an extremely local movement.

**15.2. Scope ambiguities: Type lifting?**

Next let us take up the question of whether we need QR for resolving scope ambiguities. Consider the ambiguous sentence:

(290) **Everybody likes somebody**

The reading with wide scope of the subject with respect to the object can be obtained by assuming a type-lifting rule as before, i.e. by giving the verb *likes* the type ((et)(et)). The lexical entry is formulated in strict analogy to the one given before. The analysis would be as before with the difference that the meaning of the subject has to be applied to the meaning of the VP when we evaluate the structure. To generate the reading where the object has wide scope with respect to the subject, we need a second lexical entry:

(291) Another type-lifting of *like*  
\[ F(like_{((et)(et)(et)))) = \lambda w.\lambda P_{(et)}.\lambda Q_{(et)}.P(\lambda x. Q(\lambda y. y \text{ likes } x \text{ in } w)) \]

As the reader may check for himself, the same SS gives us the intended reading:

(292) \[ [t \ \text{everybody} [((et)(et)(et))] \text{ likes } [\text{somebody}]] \]
\[ \lambda w.(\exists x)[\text{person}(x,w) \land (\forall x)[\text{person}(y,w) \implies \text{likes}(y,x,w)]] \]

We have to assume an ambiguous entry for the verb *like*. But QR is eliminated from the syntax. However, rule (291) overgenerates for it gives *everybody* wide scope over a negative quantifier in subject position:

(293) \[ *[t \ \text{nobody} [((et)(et)(et))] \text{ likes } [\text{everybody}]] \]
\[ \lambda w.(\forall y)[\text{person}(x,w) \implies \neg(\exists y)[\text{person}(y,w) \land \text{likes}(y,x,w)]] \]

This reading (“Nobody likes anybody”) is not possible for the sentence. So we have to block it. We would have to say that the entry in (291) is defined only for positive quantifiers. This leaves us with the problem to say what this means. In view of what we have said about negative indefinites, the representation in (293) is not correct. A better syntax would be:
(294) Op., nobody likes everybody

Here nobody means “somebody”. An appropriate lifted verb meaning would then give us the reading: “Not everybody likes somebody.” It is not clear how this could be blocked. There are other methods to get rid of QR by type lifting operations (see e.g. (Barker, 2004)), but they encounter similar problems.

Another example requiring an application of QR that cannot be eliminated in an obvious way by a type-lifting rule is the relative reading of the superlative statement (123), i.e., John climbed the highest mountain. Recall that we had to represent this as:

(295) John ESTc λd [climbed [THE [NP[d highest] mountain]]]

The superlative operator has to be QR-ed to an adjunction position of the VP containing it. Since it originates in an adjective located inside a quantifier, I see no way to express the reading by somehow manipulating the entry of the verb climbed. I conclude that we must have QR for resolving scope ambiguities. So we have to speak about restrictions.

15.3. Clause-Boundedness?

Here is a case of long QR that produces a non-existing reading:

(296) Bill invited a girl every student likes.
*every student λ3 Bill invited a girl t3 likes
  = λw.(∀x)(x is a student in w → Bill invited in w a girl in w that x likes in w)

The reading in (296) is not possible. The quantifier is extracted from a relative clause and violates the Complex-NP-Constraint discovered by J.R. Ross; cf. (Ross, 1986). A way of accounting for this and other cases is to say that QR out of a finite clause is not possible (Clause Boundedness). This is folklore wisdom, presumably originating in R. May’s work.

For certain configurations the constraint seems too strict. A case in question are so-called de re readings for indefinites under attitudes, which seem to require QR:

(297) Bill believes that a girl is waiting for him.
    a girl λ4 Bill believes that t4 is waiting for him
    = λw.(∃x)(x is a girl in w & (∀w’)(w’ is compatible with the beliefs of Bill in w → x
    is waiting for Bill in w’))

Note that “double indexing”, i.e. a relativization of a girl to the actual world while leaving it in the embedded clause wouldn’t be sufficient. It is important that a girl has wide scope with respect to the universal verbal quantifier believes, because Bill holds a belief about
some particular girl, which is the same in every belief world. If the standard analysis is correct, Clause Boundedness can sometimes be circumvented, and one would like to know why this is so.

15.4. Binding Theory restricts QR

Let us ask now whether we need QR for the purpose of variable binding, more precisely, the binding of pronouns. Recall first why we need QR for that purpose. The LF for the sentence No one shaved himself was this:

\[
(298) \quad \text{no one } \lambda_5 t_5 \text{ shaved himself}_5 = \lambda w. \neg (\exists x)[\text{person}(x,w) \land x \text{ shaved } x \text{ in } w]
\]

QR creates a \(\lambda\)-operator by movement and it is this \(\lambda\)-operator that binds the trace and the reflexive pronoun. The meta-linguistic translation shows the effect: the verb “shaved” has the same variable \(x\) in subject and object position, and the quantifier is moved away. So movement seems essential for the bound interpretation.

But QR may generate unwarranted readings as in the following example, which is taken from (Heim and Kratzer, 1998). The authors of this work will be called H & K in this section.

\[
(299) \quad \begin{align*}
\text{a. } & \text{The shark next to him}_1 \text{ attacked [every diver]}_1 \quad \text{(SS)} \\
\text{b. } & *[\text{every diver}]\lambda_1 \text{ the shark next to him}_1 \text{ attacked } t_1 \quad \text{(LF)} \\
& = \lambda w. (\forall x)[x \text{ is a diver in } w \rightarrow \text{the shark in } w \text{ next to } x \text{ attacked } x \text{ in } w]
\end{align*}
\]

The sentence doesn’t have this reading. And the LF that expresses it exhibits a so-called Weak Crossover configuration. The sentence in (299a) can’t have the reading in (299b). Note that QR doesn’t violate Clause-Boundedness here. The binding is simply illegitimate, and we have to block it.

There are other illegitimate pronoun bindings that are not excluded by Clause-Boundedness but by Chomsky’s Binding Theory:

\[
(300) \quad \begin{align*}
\text{a. } & *\text{John } \lambda_1 t_1 \text{ thinks that Mary hates himself}_1. \\
\text{b. } & *\text{John}_1 \lambda_1 t_1 \text{ hates him}_1.
\end{align*}
\]

QR (of the subject) is very local in these cases. We will see that it violates Principle A and B of Chomsky’s Binding Theory. Let us shortly sketch that theory.

The following examples illustrate violations of Chomsky’s Binding Principles A, B, and C. The violations are indicated by the exclamation mark “!”.
Principle A entails that a reflexive (or reciprocal) pronoun must be A-bound in an appropriate local domain (say, the next finite sentence; the actual definitions of the GB Theory are more complicated). Principle B says that a non-reflexive pronoun must be A-free in that local domain. Principle C says that a name is A-free everywhere. An NP or DP $\alpha$ $A$-binds an expression $\beta$ iff (i) $\alpha$ occurs in an argument position; (ii) $\alpha$ and $\beta$ are co-indexed; (iii) $\alpha$ c-commands $\beta$, and (iv) there is no $\gamma$ c-commanded by $\alpha$ and c-commanding $\beta$ such that $\gamma$ would bind $\beta$ as well. An expression is $A$-free if it is not A-bound. Argument positions are those of type e (and perhaps of type t).

Chomsky’s Binding Theory doesn’t interpret the structures under consideration semantically. But clearly the ungrammaticality of the examples is based on semantic intuitions: we have certain interpretations in mind that we do not accept. Let us call Chomsky’s notion of binding syntactic binding. H&K relate this notion to the notion of semantic binding. $\alpha$ binds $\beta$ semantically if $\alpha$ has created a $\lambda$-operator $\lambda x$ by QR (or another movement rule) and $\lambda x$ binds $\beta$. In other words, $\beta$ has to be an occurrence of the variable $x$ such that $\lambda x$ is the nearest $\lambda$-operator that can bind $x$. H&K state the following interface principle:

(302) Binding Principle (BP)

Let $\beta$ be a phonetically non-empty DP. Then $\alpha$ semantically binds $\beta$ iff $\alpha$ syntactically binds $\beta$.

By the Binding Principle the LFs for the structures in (301) are these:

(303) a. $\text{John}_1 \lambda_1 t_1 \text{ thinks that Mary hates himself}_1$.

b. $\text{John}_1 \lambda_1 t_1 \text{ hates him}_1$.

The binding index attached to $\text{John}$ means nothing and is therefore deleted at LF. The LFs are semantically perfectly OK. This shows that the BP cannot replace Chomsky’s Binding Conditions A, B, and C. BP establishes a correspondence of syntactic binding and semantic binding and thereby delivers an interpretation for Chomsky’s structures. We assume that A and B apply at SS. Principle A rules out (301a)/(303a) and B rules out (301b)/(303b).

It is not clear what the interpretation of (301c) should be. What could it mean to bind a
proper name? In the syntactic literature we find talk about “co-reference”. (Büring, 2005) tries to make the idea precise by considering names as operators that restrict variable assignments. I will not discuss that here. We leave the semantic status of Principle C violations unresolved and ignore that principle in the further discussion.

An immediate consequence of BP is that semantic binding is only possible iff the binding DP c-commands the pronouns binds. (This is so because syntactic binding presupposes c-command, and we have a 1-to-1 correspondence between syntactic and semantic binding.) The fact that bound anaphora is possible only under c-command is what (Reinhart, 1983: p. 122f.) has called the Bound Anaphora Condition.

The BP enforces applications of QR that are semantically unnecessary though innocuous. The following example is taken from (Heim and Kratzer, 1998: p. 264):

(304)  a. She1 said she1 saw a shark next to her1 (SS)
       b. She1 λ1 t1 said she1 λ1 t1 saw a shark next to her1 (LF)

Together with H&K’s Binding Principle Chomsky’s Binding Principles thus constrain QR because they prevent an application of QR if it entails a violation of Chomsky’s Binding Theory, and they enforce an application of QR if Chomsky’s Binding Principles want to have it that way.

H&K’s Binding Principle constrain QR such that the rule cannot generate a weak crossover constellation anymore:

(305)  a. The shark next to him1 attacked [every diver]1 (SS)
       b. *[every diver] λ1 the shark next to him1 attacked t1 (LF)

The LF is blocked by BP because every diver binds him1 semantically but not syntactically. The BP thus implies that semantic binding is only possible if the binding DP α c-commands the bindee β at SS. This would rule out a treatment of so-called Inverse Linking structures by means of QR:

(306)  a. Someone from [every city]1 despises it1 (SS)
       b. [every city] λ1 someone from t1 despises it1 LF: BP!

The LF in (306b) expresses the intended reading and it certainly is the most straightforward analysis. But it is ruled out by BP.

On the other hand, BP does not bar an LF like that in (297b). This is so because BP doesn’t speak about traces, which are phonetically empty. To be sure, the LF violated Clause-Boundedness, but it doesn’t violate the Binding Principles.
15.5. Binding Theory without QR?

If we look at the cases of legitimate pronoun binding by QR, we see that the movement is so local that we may ask whether we can get rid of QR for the purpose of binding. (Büring, 2005) offers a method of variable binding without QR. He proposes a surface interpretation of Chomsky’s indexed structures. He first formalizes the binding indices n of NPs that are not operators (i.e. no wh-phrases, relative pronouns etc.) by a binding prefix βn. The following LF-rule does this:

(307) Binder rule (Büring)

\[ [\text{NP}_n \varphi] \Rightarrow [\text{NP} [\beta_n \varphi]] \]

This almost identical to H&K’s convention for QR, where the movement index is adjoined to the XP to be modified and interpreted as a λ-operator. The difference is that the binding index n is not created by movement. It might be thought to be present at SS as Chomsky assumed or it might be created by the Binder rule itself as Büring assumes. Our version of the Binder rule sticks closer to the GB-notation. Büring calls βn binding prefix. βn is interpreted as a sort of λ-operator as well.

(308) Binder index evaluation rule (BIER)

Let \( \varphi \) be of type et. Then \([\beta_n \varphi]\) is of the same type.

\[ [[\beta_n \varphi]]^g = \lambda x.([[\varphi]]^{g[\alpha/x]}(x)) \]

Here is a simple example illustrating the rule:

(309) everyone [\beta_n shaved himselfn ]

The structure can be interpreted without QR, and it has the correct meaning. A closer inspection of principle BIER reveals that it is a lexicalised version of QR: βn is not simply a λ-operator. It binds the variable \( n \) that occurs in the VP \( \gamma \) and it introduces the same variable as a subject of the VP and identifies it with the interpretation of \( n \). If we apply a generalised quantifier to that meaning, we get exactly the same result as if the quantifier had been QR-ed out of the subject position of the sentence.

Büring’s method can be applied to objects that bind variables, but we need a more complicated composition principle for this case:

(310) a. Hans erklärte jedem1, dass er1 inkompetent sei.

Hans told everyone1 that he1 incompetent was.
b. Hans [et everyone [[(et)t] t] β₁ [et] told [t that he₁ incompetent was]]]

The rule for binding prefixes that bind a dative position is this:

(311) **BIER for the indirect object**

Let γ be of type (e(et)). Then [βₙ γ] is of type ((et)t)(et).

\[[ [βₙ γ] ] \] = \[λ_κ(e(et))·λ_κQ(λ_κy( [ [γ] ] )^ₙ(x))\]

As the meta-linguistic notation shows a local version of QR is hidden in this rule. But it provides the correct interpretation as the reader may calculate for herself.

Obviously, there is a general method behind the different binding rules. We may generalize the compositional principles to get some general scheme. See (Büring, 2005) for details. For binding purposes this method seems to work rather well and it provides a viable alternative to block overgeneration.

Let us see how Büring treats Crossover. Consider example (299) again:

(312) **The shark next to him₁ attacked [every diver]ₙ** (SS)

The index of the quantifier at object position makes little sense because there is no pronoun in the c-command domain of the object. So crossover cannot arise. The rule that combines the object with the verb is stated in analogy to the two rules given and left to reader.

The standard cases of crossover are however those that arise when a wh-phrase binds a pronoun through movement as in the following examples:

(313) **Weak Crossover**

\[\text{Whom₁ does his₁ mother hate t₁ ?} \]

≠ Which person is hated by his mother?

Note that the corresponding German sentence is grammatical; therefore weak crossover shouldn’t follow from the binding formalism without an additional principle. Büring proposes to treat crossover by separating traces created by moved wh-phrases from pronouns. Pronouns like heₙ or himselfₙ are variables represented by pairs of natural numbers n and type e, written as ne. Traces are represented by pairs <t,n> consisting of the letter t and the number n, written as tn. Call the former pronoun variables and the latter trace variables. Both kinds of variables are assigned individuals by an assignment function g, but g(nₙ) may be different from g(tₙ). Büring represents movement indices of wh-phrases by μ₁. The rule that interprets movement is this:

(314) **The movement interpretation rule (Büring)**
A moved wh-phrase $\alpha$ of type (et)t has the index $\mu_i$. The trace is $t_i$, i.e. we have co-indexing. Let $\beta$ have the type t.

$$[[\alpha[\mu_i \beta]]] = \lambda w. [[\alpha]](\lambda x.[[\beta]] g^{ui(x)})$$

The difference from the standard interpretation of QR is that the abstraction only binds trace variables but not pronoun variables. The interpretation of the question in (313) is this:

(315) $\text{who } \mu_n \text{ does his mother hate } t_n$?

$$= \lambda w. \lambda p. (\exists x) [\text{person}(x,w) \& p = \lambda w'. g(\text{he}_1)'s \text{ mother in } w' \text{ hates } x \text{ in } w' \& p(w)]$$

We see that the possessive pronoun is free. It denotes a person delivered by the context $g$. So no crossover arises. Nevertheless, there is a way to produce the unwanted bound reading, as noticed by Büring:

(316) $\text{who } [\beta_n[\mu_n \text{ does his mother hate } t_n]]$?

The reader may convince himself that this LF indeed represents the bound reading “Who is hated by his own mother?” To block this LF, Büring (p. 169) imposes the restriction to the Binder rule (307) that the NP must not have undergone wh-movement.

Note that earlier approaches in Generative Grammar differ from Büring’s proposal in so far as they try to exclude the structure in (313) as ungrammatical. For Büring the structure is well formed. The interpretation of the movement rule makes sure that the pronoun cannot be bound by the wh-phrase.

Strong crossover as exhibited in the next example is explained in the same way:

(317) Strong Crossover

Whom$_1$ does he$_1$ admire $t_1$?

$\neq$ Who admires himself?

Strong Crossover arises when a wh-phrase semantically binds a pronoun that A-binds a wh-trace. This however is not possible if a wh-phrase cannot semantically bind a pronoun.

As remarked above, German doesn’t have the weak crossover effect, but it shows the strong crossover effect. Here are the translations of the two relevant sentences:

(318) a. OK $\text{Wen}_1 \text{ hasst seine}_1 \text{ Mutter}$?

b. $\ast \text{Wen}_1 \text{ bewundert er}_1 \text{ ti}_1 \text{ ti}_1$?

(The acceptability judgements refer to the bound readings of course.) It is unclear what the best way is to derive these readings. We have to say that German admits the LF (316) but it
doesn’t allow for the corresponding strong crossover configuration

(319)  
\[ *\text{who}_1 \beta_1 \mu_1 \text{he}_1 \text{admires}_1 t_1 \]

Chomsky’s GB-theory bars this structure by Principle C. Wh-traces count as “r-expressions” (referential expressions like names) and must not be A-bound. Something like this is needed in Büring’s theory as well.

15.6. Conclusion

It is possible to eliminate QR by means of appropriate type lifting rules when we are dealing with quantifiers in object position. The elimination amounts, however, to hide QR in the semantic meta-language. It seems harder to eliminate QR from the syntax when we are dealing with scope ambiguities. In certain cases this doesn’t seem to be possible at all, e.g. in superlative constructions. The same point can be made for comparative constructions. Variable binding requires QR. We have seen that the Binding Theory and the Binding Principle constrain the application of the rule. If we assume Büring’s version of the Binding Theory, we appear to have eliminated QR from the syntax. But the elimination is spurious. We need a number of rules that interpret the binding index. Each such rule may be regarded as a locally constrained application of QR in the syntax. My conclusion is that we should keep QR in the grammar. It is a central rule of construal, but its application has to be restricted. I don’t know what the best way of restricting the application is. The most attractive idea is a restriction by economy: apply QR as much as the interpretation requires (and not more). For an idea of this kind, see (Fox, 2000).

A non-note on the literature. The type lifting methods are used in generalised categorial grammar. See (Szabolcsi, 1989), (Jacobson, 1999), (Steedman, 2000), (Barker, 2004) among many others. A very good discussion of the different methods of dealing with quantifiers is found in (Heim and Kratzer, 1998: 7.5). My discussion of the Binding Theory is based on only two sources, viz. (Heim and Kratzer, 1998) and (Büring, 2005). The latter work contains a careful discussion of different approaches to the Binding Theory, and the reader should consult it.

16. DONKEY PRONOUNS: BEYOND QR?

This section treats so-called donkey sentences. These contain pronouns that seem to be bound but cannot be bound by QR. Thus these sentences seem to require a different syntax and semantics from what we have been assuming in this article. I will discuss the Kamp/Heim
approach to these constructions. Recently, another method to deal with the problem has become popular, the method of the E-type pronouns. This method has to use a special version of possible world semantics, viz. situation semantics. The approach will leave donkey pronouns free and is compatible with the architecture of grammar we have been assuming so far.

16.1. The Binding Problem for Donkey Pronouns

(Geach, 1962) has brought our attention to sentences that contain bound pronouns that cannot be analysed by means of QR:

\[(320)\]
\[
\begin{align*}
\text{a. If a farmer owns a donkey, he beats it.} \\
\text{b. Every farmer who owns a donkey beats it.}
\end{align*}
\]

In (320a), the pronoun he seems to be bound by a farmer, and in (320a/b), it seems to be bound by a donkey. Let us try to bind the pronouns of the first sentence by means of QR. The only way to do this consists of giving a farmer and a donkey scope over the entire conditional, i.e., the LF has to be something like the following construction:

\[(321)\]
\[
\text{a farmer } \lambda_1 \text{ a donkey } \lambda_2 \text{ if } t_1 \text{ owns } t_2, \text{ he}_1 \text{ beats it}_2
\]
\[= \text{There is a farmer } x_1 \text{ and a donkey } x_2 \text{ such that if } x_1 \text{ owns } x_2, \text{ then } x_1 \text{ beats } x_2.\]

This LF speaks about a particular farmer and a particular donkey and says that the farmer beats this donkey in case he owns it. Even if there exists this reading, which is doubtful, it certainly isn’t the prevailing interpretation. Sentence (320a) is synonymous with sentence (320b) and both seem to have a universal reading, something like:

\[(322)\]
\[
\text{For every } x \text{ and } y, \text{ if } x \text{ is a farmer and } y \text{ is a donkey and } x \text{ owns } y, \text{ then } x \text{ beats } y.
\]

The problem raised by these “donkey sentences” is then how we can obtain this reading in a compositional way from surface syntax.

The binding problem for donkey pronouns can be resumed like this. Semantic binding is only possible if the antecedent c-commands the pronoun to be bound. Donkey pronouns seem to have antecedents binding them. But the antecedents don’t c-command the pronouns.

16.2. Unselective Binder Approaches

The classical solutions of the problem are due to (Kamp, 1981) and (Heim, 1982). They are inspired by (Lewis, 1975) who assumes that a farmer and a donkey are not existential quantifiers but rather open propositions of the form farmer(x) and donkey(y); the donkey pronouns are interpreted as variables x and y as usually. The quantification is done by an
unselective binder, which universally quantifies over the free variables in the antecedent and the consequent of the conditional. Lewis’ idea is that the source of the quantification is an adverb like always, sometimes, never, often, seldom,… The if-clause restricts the quantifier. Thus, the LF of a donkey sentence is something like this:

(323)

\[
\{ \begin{array}{l}
\text{always} \\
\text{sometimes} \\
\text{never} \\
\text{often} \\
\text{seldom}
\end{array} \] \text{ if a farmer owns a donkey, he beats it.}
\]

The adverb always may be covert. Let us represent it by capitals. Assuming that if means nothing, the most straightforward semantics for ALWAYS would then be the following:

(324) \[ \begin{array}{l}
\text{ALWAYS } \varphi, \psi \strut \\
\text{iff for every } g' =_{F(\varphi)} g \text{ such that } \varphi^g = 1, \psi^g = 1.
\end{array} \]

(preliminary)

\( g' =_{F(\varphi)} g \) means that the assignments \( g' \) and \( g \) coincide in their values with the possible exception of the variables in \( \varphi \) that are free for binding. These are precisely the variables that are introduced by indefinite terms, where each indefinite term introduces a new variable (Heim’s novelty condition). With respect to the entire conditional, the variables are bound, of course.

As it stands, rule (324) is too simple because the consequent may contain indefinite terms, and these must be existentially quantified while the variables (i.e. pronouns) free for binding must be bound by ALWAYS. As an example, consider the following sentence:

(325) a. If a farmer owns a donkey, he buys it a saddle.
   b. For every \( x \) and \( y \), if \( x \) is a farmer and \( y \) is a donkey and \( x \) owns \( y \), then there is a \( z \), \( z \) is a saddle and \( x \) buys \( y \) \( z \).

Examples like this motivate the following revision of the ALWAYS-rule.

(326) \text{The ALWAYS-rule}

If \( \varphi \) and \( \psi \) are expressions of type \( t \), then [ALWAYS if \( \varphi, \psi \)] is of type \( t \) as well.

\[ \begin{array}{l}
[\text{ALWAYS } \varphi, \psi] = 1 \text{ iff } \forall g' \{} g' =_{F(\varphi)} g \text{ & } \varphi^g = 1 \rightarrow \exists g'\' \{} g'\' =_{F(\varphi)} g' \text{ & } \psi^g = 1\}
\end{array} \]

Suppose we manage to build up the following LF for the sentence in (325):

(327) \[ \text{if [a farmer } x \text{ and a donkey } y \text{ and } x \text{ owns } y ] [a saddle } z \text{ and } he \text{, buys it, } z \text{ ]} \]
If we can make sure that \( x \) and \( y \) are the variables free for binding in the antecedent and \( z \) is the variable free for binding in the consequent, the ALWAYS-rule will give us precisely the reading stated in (325b).

The account given is not yet complete, of course. We have to say how the LFs containing indefinite terms are obtained from surface syntax. (Heim, 1982), in the first part, treats indefinite terms as open propositions containing new variables. They are scoped out of their base position leaving a co-indexed definite trace. Thus we obtain two (or more) open propositions that are interpreted conjunctively (“cumulatively”). The antecedent in of the construction in (327) is therefore more precisely something like:

\[(328) \quad \text{a farmer}^{1i} [\text{a donkey}^{2i} [t_1^d \text{owns} t_2^d]]\]

The superscripts \( i = \text{indefinite} \) and \( d = \text{definite} \) reminds of the different status of the variables. Indefinite variables are new, definite variables are old. Traces are old.

This process of building cumulative formulas is rather similar to QR. The binding of the variables is then done by the ALWAYS-rule. Another problem not discussed here is the precise definition of the syntactic concept “free for binding”.

The theory of Kamp is similar in spirit but technically somewhat different. The logical language used is called DRT (discourse representation theory). The expressions of the language are called DRS (discourse representation structure). A DRS consists of conditions, which are similar to the parts of the cumulative formula in (328). Each DRS is prefixed by a list of all the variables that are free for binding with respect to the DRS, the universe of the DRS. In Kamp’s theory, the donkey sentence in (325a) may be written as\(^{13}\):

\[(329) \quad \text{ALWAYS} [x, y : \text{farmer}(x), \text{donkey}(y), x \text{ owns} y] [z : \text{saddle}(z), x \text{ buy} y z]\]

The semantics is as before with a proviso I haven’t mentioned yet. Both Kamp and Heim work with partial assignments, i.e., assignments that are only defined for the variables that are used in a text, viz. a DRS. For instance, the meaning of the antecedent in (329) is the set of those partial assignments that satisfy the three conditions listed in the DRS representing the antecedent. A restatement of the ALWAYS-rule for Kamp’s system is therefore the following:

\[(330) \quad \text{ALWAYS in DRT}\]

\(^{13}\) (Kamp and Reyle, 1993) uses the symbol \( \Rightarrow \) for ALWAYS.
For any assignment $g$: $g$ satisfies the DRS $[\text{ALWAYS } \varphi, \psi]$ iff $\forall f_1 [\text{If } g \subseteq f_1 \& f_1$ satisfies $\varphi \rightarrow \exists f_2 [f_1 \subseteq f_2 \& f_2$ satisfies $\psi]]$

Indefinite pronouns can thus be bound by the ALWAYS-rule. The semantics generalizes to the other adverbs of quantification. For instance, \textit{often} has the semantics “for many assignments $f_1$: …”. It is known that quantifiers of this sort cannot be expressed in first order logic; cf. (Barwise and Cooper, 1981). Negation is an unselective variable binder, too. For instance, the sentence \textit{Nobody laughs} would have the following LF:

\hspace{1cm} (331) \hspace{1cm} \text{NEG } [\text{somebody } 1^i [t_1^d \text{ laughs}]]

And NEG would be a binder of the free variables:

\hspace{1cm} (332) \hspace{1cm} \text{The NEG-rule for Kamp/Heim-systems} \\
\hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} [\text{NEG } \varphi]^g = 1 \text{ iff there is no } g': g' =_{F(\varphi)} g \& [\text{NEG } \varphi]^g$.

We thus need an abstract negation for negative indefinites, which goes together well with Penka’s account of negation. But the semantics is rather different: the negation is an unselective negative universal quantifier. The systems of Kamp and Heim are made for anaphoric relations in discourse. The indefinite variables introduced by indefinite terms can be taken up anaphorically by a definite pronoun but they can never refer anaphorically themselves. A way of achieving this is to make sure that an indefinite term always introduces a new variable (Heim’s novelty condition) while a definite term is an old variable (Heim’s familiarity condition). So the discourse in (333a) has the LF in (333b).

\hspace{1cm} (333) \hspace{1cm} a. \hspace{0.2cm} A farmer owned a donkey. He bought it a saddle. \\
\hspace{1cm} \hspace{1cm} b. \hspace{0.2cm} [a \hspace{0.2cm} \text{farmer } 1^i [a \hspace{0.2cm} \text{donkey } 2^i [t_1^d \text{ owns } t_2^d ]] [a \hspace{0.2cm} \text{saddle } 3^i [he_1^d \text{ bought } it_2^d t_3^d]] \\
\hspace{1cm} \hspace{1cm} c. \hspace{0.2cm} (\exists x_1)(\exists x_2)(\exists x_3)[\text{farmer}(x_1) \& \text{donkey}(x_2) \& x_1 \text{ owns } x_2 \& \text{saddle}(x_3) \& x_1 \text{ bought } x_3 \text{ for } x_2]

The sentences of a discourse are interpreted conjunctively. On the discourse level, the free variables are existentially bound. This is done by an appropriate definition of the notion of truth of a discourse. This method will give us the truth-condition in (333c).

The method for interpreting conditionals looks rather attractive especially in view of a missing alternative given the methods introduced so far. The procedure becomes suspiciously ugly, however, if we turn to the “relative donkey”-construction.

\hspace{1cm} (334)
In the Kamp-Heim-theory, the LF of this sentence must be exactly alike to that of the sentences in (323). For instance, the *every*-sentence must have the LF in (329). In order to achieve that, we have to make the rule for relative clauses such that the relative clause is interpreted conjunctively with its head noun. Furthermore, the determiner must be moved out of the DP getting scope over the entire construction without leaving a trace. It functions then analogously to an adverb of quantification. I won’t elaborate the rather complicated details but I refer the reader to the first part of Heim’s dissertation (Heim, 1982). Obviously this treatment of the determiners does not go together with the syntax of natural language; all the useful work QR does in semantics and for the interface would have to be replaced by syncategorematic ad hoc procedures.

But even empirically the syncategorematic approach to donkey anaphora is not without problems. There is the problem of asymmetric readings first noted by (Bäuerle and Egli, 1985):

(335) Most farmers that have a donkey beat it.

Suppose we have ten farmers. One of them has hundred donkeys and beats each of them. The other nine farmers own one donkey each and treat it well. Intuitively, the sentence is false in this scenario. The truth-condition given by the Kamp-Heim-theory predicts truth, because we have 109 farmer-donkey-pairs satisfying the restriction of the quantifier, and 100 thereof satisfy the nuclear scope. This is more than half of the pairs.

Another problematic example, discovered by (Pelletier and Schubert, 1989), is that of weak readings.

(336) a. Everyone who had a dime put it in the meter.
    b. If you have a dime, put in the meter!

Sentence (336a) doesn’t mean that everyone put all of his dimes into the meter. It rather means that everyone put one of his dimes in the meter. I don’t know whether there is a convincing solution to the latter two problems. But there is an analysis of donkey pronouns compatible with our previous architecture and with the central role of QR, namely the E-type approach to donkey pronouns.
16.3. The E-Type Approach

The E-Type approach to donkey anaphora has its origins in (Evans, 1980). A precise technical implementation is due to (Cooper, 1983). The method has become popular again through (Berman, 1987) and (Heim, 1990). The approach outlined in this section is basically that of (Elbourne, 2005). The idea is that E-type pronouns\(^{14}\) stand for descriptions whose restriction is to be recovered from the context. The paraphrase for a donkey sentence is then something like this:

(337) If a farmer has a donkey, the farmer that has the donkey beats the donkey he has.

The paraphrase sounds good, but it needs some appropriate theoretical assumptions. As it stands, the sentence seems to presuppose that there is only one farmer that owns a donkey in the whole world. Similarly for the donkey owned. But we know that the natural interpretation is a universal statement about farmers and donkeys. The way out is quantification about minimal situations along the following lines:

(338) Every minimal situation containing a farmer that owns a donkey is a situation in which the farmer beats the donkey.

To make this precise, we introduce the following assumptions about the ontology of situations.

(339) 1. *Situations* are parts of worlds. They are ordered by the mereological part-of relation $\leq$. As usual, $<$ is the proper part-of relation. For every situation $s$ there is a maximal situation $w(s)$, the world of $s$.

2. Propositions (sentence-intensions) are partial functions from situations to truth-values.

3. A situation $s$ satisfies the proposition $p$ if $p(s) = 1$. $s$ minimally satisfies $p$ — Min($s,p$) — iff $s$ satisfies $p$ and no proper part of $s$ satisfies $p$.

4. All natural propositions (and relations) are persistent, i.e., if a situation $s$ satisfies $p$, then every bigger situation satisfies $p$ as well (similarly for relations).

To make quantifiers work in situation semantics, we have to relativize them to minimal situations satisfying them. Here are the quantifiers we need for simple donkey sentences:

(340) The indefinite article in situation semantics

---

\(^{14}\) I don’t know what the letter “E” stands for. Perhaps it stands for “Evans”.

115
a has the type (s(et))(s(et)t)
\[ F(a) = \lambda s. \lambda P_{s(et)}. \lambda Q_{s(et)}. (\exists x)(\exists s')[[s' \leq s \land \text{Min}(s', \lambda s.P(s)(x))] \land (\exists s'')[s' \leq s'' \land \text{Min}(s'', \lambda s.Q(s)(x))]] \]

(341) **ALWAYS** in situation semantics

has the type (st)((st)t).
\[ F(\text{ALWAYS}) = \lambda s. \lambda p. \lambda q. (\forall s' \leq s)[\text{Min}(s', p) \rightarrow (\exists s'')[s' \leq s'' \land \text{Min}(s'', q)]] \]

The last ingredient we need is the donkey pronoun, which denotes the definite article with a restriction determined by the context.

(342) **E-type pronouns** (type (s(et))e)
\[ F(\text{he/she/it}) = \lambda s. \lambda P_{s(et)}. (\exists! x) \text{Min}(s, P(s)(x)). \text{tx}. \text{Min}(s, P(s)(x)) \]

The LF for our sentence is now this:

(343)

The symbol \(^v\) is Montague’s extensor or down-operator, which has the following semantics:

(344) **Montague’s down-operator**

Let \(\alpha\) be of type sa. Then \(^v\alpha\) is of type a. \(\[
\][\]
\[\lambda s. ([\]
\][\]) (s)(s) \]

I will comment in a moment why we need this operator. Let us assume the following values for variables 1 and 2 respectively:

\[ g(1) = \lambda s. \lambda x. x \text{ is a farmer in } s \]
\[ g(2) = \lambda s. \lambda x. x \text{ is a donkey in } s \]

Under these assumptions the truth-condition of the sentence is the following:

\[ \lambda s. (\forall s')[s' \leq s \land \text{Min}(s', \text{that a farmer has a donkey}) \rightarrow (\exists s'')[s' \leq s'' \land \text{Min}(s'', \text{the g(1) beats the g(2)})]] \]
We see that the donkey pronouns are free variables. We would expect that their content is determined by their intuitive antecedents. This however is not so. The pronouns give us properties and they have to be somehow inferred from the sentence. That is the reason why (Heim, 1990) has a copy procedure that copies the content of these pronouns somehow from the antecedent. We leave the theory as it stands, however.

Here is the comment on the necessity of the addition of Montague’s extensor for the interpretation of the restriction of the E-type pronouns. The reason is that variables have a rigid interpretation. Suppose we chose a simpler approach that assumes the type et for the two variables. Then g(1) would be a particular set of farmers that couldn’t co-vary with situations. On the other hand, we could try to omit the extensor and work with FA. Then the presupposition of the donkey pronoun 1 would be \( \lambda s.(\exists x!x)\lambda s'.x \) is a farmer in s’. This is an undefined expression. The extensor in front of the restriction gives the correct result, as the reader may calculate for himself.

The “relative donkey” requires E-type pronouns containing a variable bound by the subject. This can be seen from an inspection of the truth-condition of the sentence in (320b):

(345) Every farmer that owns a donkey beats it.

\[
\lambda s.(\forall s')(\forall x)[s' \subseteq s \& \text{Min}(s', \text{that } x \text{ is a farmer that owns a donkey}) \rightarrow x \text{ beats the donkey that } x \text{ owns in } s']
\]

The two-place relation ‘being a donkey owned by’ is recovered from the context. The subject variable x in the consequent is created by QR, and the second variable is an anaphoric pronoun contained in the restriction of the E-type pronoun. Thus the LF is the following tree:

(346)
The value assumed for the variable \( 5 \) is \( \lambda s.\lambda x.\lambda y. y \) is a donkey owned by \( x \). Again, this pronoun is not bound by the antecedent, but it contains a variable bound by the antecedent, viz pro1/x.

The only thing still needed is the accommodation of every to this analysis: it must quantify both over individuals and situations:

\[
(347) \quad \text{every} \quad \text{in situation semantics:}
\]

\[
[[\text{every}]] = \lambda s.\lambda P_{s(e(t))}.\lambda Q_{s(e(t))}.(\forall x)(\forall s')[[s' \leq s \& \text{Min}(s', \lambda s.\lambda P(s)(x))]\rightarrow (\exists s'')[s' \leq s'' \& \text{Min}(s'', \lambda s.\lambda Q(s)(x))]]
\]

The determiner some has an analogue interpretation. The analysis of the relative pronoun is standard, i.e., it is the semantically empty pronoun. The elaboration of the LF for the relative clause is left to the reader. He may convince himself that the LF yields the truth-condition given in (345).

There is much more to be said about these constructions, but we leave it at this stage. If the approach is on the right track, donkey pronouns present no obstacle for QR because they are free variables.

16.4. Notes on the Literature

The classical references for the theory of donkey sentences are (Kamp, 1981) and (Heim, 1982). A very good and short presentation of Heim’s theory is (Haas-Spohn, 1991). The standard textbook for Kamp’s theory is (Kamp and Reyle, 1993). The approach to E-Type pronouns is found in (Heim and Kratzer, 1998) as well. The situation theory used in this section originates with (Kratzer, 1989) as far as I know. The theory isn’t complete as it is...
presented here. It doesn’t work for statives. Presumably there is no temporally minimal situation that satisfies “Mary be sick”. Among other things, this problem is addressed in (Kratzer, 2006).

17. SUMMARY

This article has presented a theory of semantic interpretation that fits a grammar of the GB-type. I have gone through a number of non-trivial constructions of natural language to see what is needed for this purpose. Here is the repetition of some of the results.

The basic conception is that syntax generates triples of structures ($\pi$, $\rho$, $\sigma$) where $\pi$ is a phonetic interpretation of the sentence $\rho$ and $\sigma$ and is a semantic interpretation of $\rho$. $\pi$ is the structure we “hear” (or “see”, if we take writing for granted). Of course, we can only hear the terminal string. So we can do more than our senses can grasp. $\rho$ can be quite different from $\pi$ and $\sigma$ even more so. We have been assuming throughout that one LF represents precisely one meaning of a sentence. We have assumed that $\rho$ is an S-structure. So the task was to transform S-structures into LFs. The latter are parts of a $\lambda$-categorial language. And these languages have a precise interpretation, i.e., each structure is mapped to a proposition (or context-dependent proposition).

Some semanticists believe that the choice of the semantic language is arbitrary. I think this is not so. There seems to be good evidence that $\lambda$-languages mimic natural languages. If we accept the H & K theory of semantically empty pronouns PRO, we have a device that generates $\lambda$-operators, i.e., variable binders.

When we translate SSs into LFs, two cases may arise: a word might have no lexical meaning. It is there for purely syntactic reasons, e.g. the “genitive” preposition of, the complementizer that and others. Words like these are deleted by the Principle of Full interpretation ($FI$).

Or it might be that a word is not pronounced, i.e. phonetically covert. We have seen a couple of these: the tenses Present ($N$) and Past ($P$), the comparative ($ER$) and the superlative operator ($EST$), the cumulative operator $\ast$, the negation Op., the question operator ?. All of these operators are made visible by an appropriate morphology: tenses are made visible by the tense morphology of finite verbs, the comparative and the superlative have a reflex in the adjective morphology, semantic negation has reflexes in negative indefinites and so on. To understand the licensing conditions for the corresponding morphology is crucial for an understanding of natural language semantics. The semantic operators that license the
morphology can be quite distant from the forms that point to them.

The most important part of semantics is the interpretation of the syntactic rules. Our interpretation of phrase structure rules (EM) is very simple: we have FA and PM. Variable free grammars have much more rules that use very complicated types. The types of our system are very simple. In most cases it suffices to have types of second order (e.g. (et)t and (st)t, i.e. quantifiers and questions).

More interesting is the interpretation of movement rules (IM). There is movement on the surface like head movement, scrambling, raising, topicalization and perhaps other rules. If these rules do not affect the interpretation, the moved phrases are simply reconstructed at LF. (We could have interpreted the trace by leaving a variable of a higher type, but that would have made things only more complicated.) The interesting movement is the type driven one, here generally addressed as QR.

There are no obvious instances of QR that are visible at the surface. (We assumed that comparatives presented such cases. But this is not compelling: the than-clause could have been extraposed by a PF rule.) So QR is a rule that operates on the LF branch. We needed it for resolving type conflicts, for spelling out scope ambiguities, and for binding variables. I discussed proposals of getting rid of the rule, but none of these was convincing. And the interpretation of comparatives and superlatives made the existence of QR even more compelling. Some applications of QR were not entirely straightforward: in plural constructions we had to QR occasionally in a “parallel way”. The same was necessary to get certain superlative LFs right.

A hard chestnut for this QR centred view is the existence of donkey pronouns. These seemed to require syncategorematic rules of a rather different sort. If we, however, accept the theory of E-type pronouns, donkey sentences present no problems for variable binding by means of QR.

Another important interpretation of movement is operator movement, i.e., WH- and PRO-movement. Here we have a syntactic device for creating λ-operators. There are restrictions for this rule. If a semantically empty pronoun is generated in a case position, it is the relative pronoun WH, which moves to a peripheral position (Chomsky’s A-bar position). A PRO that originates in a non-case position moves extremely locally, just to form a λ-abstract. Some of these restrictions should follow from the binding theory, a question we haven’t addressed, because we were concerned with semantic interpretation.

We have said very little about PF. One point was that we assumed elliptic structures as
input when we were dealing with comparative constructions. English VP-deletion would have been another phenomenon to the point. Let me make one comment, however. If we follow Heim’s view that many features (our u-features) are transmitted by semantic binding, we seem to have a problem for the minimalist conception of phases, which was sketched in section 3. Suppose CP is a phase, i.e., the phonetic form of CP cannot be altered anymore when we have reached a later phase of the derivation. But bound variable get their u-features from their binder, and this one can be very distant from the variable. It would seem then that we must memorize the variables used until the end of the derivation.

There are many semantic phenomena of natural language I haven’t addressed in this article. For instance, I haven’t talked about deixis, about presuppositions or focus. I am pretty confident that the incorporation of deixis doesn’t change the picture essentially; the reader is referred to (Zimmermann, 1991). But presuppositions might change the picture: perhaps we have to use a dynamic framework in the style of Heim’s dissertation. Focus syntax and semantics raises interesting question for the PF/LF-interface. I refer the reader to the work of Mats Rooth, e.g. (Rooth, 1992). So there is more to say. But: iam satis est.

18. INDEX

\(\lambda\), 14, 15, 29
\(\alpha\) binds \(\beta\) semantically, 104
\(\Delta\)-interpretation, 50
(Weak) Crossover, 103
*-insertion, 87
A negative indefinite as an n-word, 90
A particle verb, 37
A-binds, 104
accessibility relation, 33
a-intensions, 28
ALWAYS in DRT, 112
ALWAYS-rule, 111
Another type-lifting of like, 101
antecedent, 11
anti-verbatimical, 99

Antonyms, 47
Argument positions, 104
at least \(n\), 60
at most \(n\), 60
Baker-sentence, 41
BECOME, 96
BIER, 106
BIER for the indirect object, 107
Binder index evaluation rule, 106
Binder rule, 106
binding prefix, 106
Binding Principle, 104
Bound Anaphora Condition, 105
Chomsky’s Binding Principles, 103
Chomsky’s Binding Theory, 103
classifier, 78
Clause Boundedness, 102
coop-distributive, 84
comparative operator ER, 45
Complex-NP-Constraint, 102
Compositionality, 4
constituent question, 39
contrast argument, 86
conversational background, 33
cover of x, 83
Cumulation Operator *, 80
Cumulation Operator *C, 83
de dicto, 33, 35
de re, 33, 34
definite article, 22, 77
degrees, 42
dependent plural, 85
donkey pronouns, 110
Double Access, 68
Double Negation (DC), 91
Double Negation (DN) languages, 92
downward entailing (DE), 98
DRS (discourse representation structure), 112
DRT (discourse representation theory), 112
EL, 13
empty pronouns, 23
extension, 27
extensional, 29
External Merge, 9
External Merge (EM), 10, 15
FA, 5, 14, 15, 29
free for binding, 111
Frege Principle, 4
French negation, 93
functional application (FA), 5
functional types, 13
fusion of the elements of M, 77
g[x/u], 15
generalised domain of individuals, 76
generalised quantifier, 18, 23
German V2-movement, 37
German verb movement, 37
H&K, 13
head, 10
head movement, 11
Heims PI-operator, 49
IFA, 28, 29
intensional, 29
Intensional Functional Application, 28
Intensional Logic, 27
intensions, 27
Internal Merge, 9
Internal Merge (IM), 10, 16
interpretation [[ . ]]^{M,s} for EL, 15
interpretation [[ . ]]^{M,s} for IL, 29
Inverse Linking, 105
L, 14, 15, 29
Less than n, 59
lexical trees, 9
lexicon, 9, 14
lifted ER-operator, 49
Logical Forms (LFs), 7
logical type, 9
logical types, 12
Main Characteristics of IL, 28
mixed predicate, 79
Modal, 33
model, 15
Montague’s ^-operator, 31
more than n, 59
Multiple Agree, 90
multiple question, 39
Negation, 90
Negative Concord (NC), 91
negative indefinite as negative quantifier, 90
Negative Polarity Items (NPIs), 98
Neg-Raising, 30
NM, 94
Non-Boolean and, 78
non-strict NC-language, 91, 92
novelty condition, 111
NPI, 50
nuclear scope, 46
numeral, 78
overlap, 76
Pair Cover, 84
Partial wh-movement, 40, 41
passive auxiliary be, 64
Past tense morphology, 62
Past with domain restriction, 70
P-atom, 77
Perfective operator PF, 72
Phonetic Forms/Representations (PFs), 7
Pied-Piping, 41
Plural Operator (for nouns), 77
PM, 14, 15, 29
Positive-operator, 57
possible worlds, 27
prejacent, 33
Principle A, 104
Principle B, 104
Principle C, 104
Principle of Full Interpretation (FI), 17
PRO, 24
problem of the object, 18
PRO-movement, 16
pronoun binding, 21
proportional quantifier, 54
Propositional attitudes, 67
purely collective predicate, 79
purely distributive predicate, 79
QR, 23
Quantifier Raising (QR), 19
Quantifying-in, 23
quantized predicate, 77
question operator ?, 39
range argument, 86
re/de dicto ambiguity, 33
reciprocal pronoun, 86
Reconstruction, 30, 33, 38
relative pronouns, 24
restriction, 45
Rules of Construal, 9
scope, 21
semantic domains, 13, 27
semantic lexicon, 14, 28
Semantic Past, 62
Semantic Present, 62
semantically reconstructed, 35
split scope of NIs, 94
strict NC languages, 92
strict NC-language, 91
Strong Crossover, 108
superlative EST, 51
surface compositionality, 7
surface structure, 7
syntactic binding, 104
syntactic category, 9
syntactic reconstruction (Rec), 35
Syntactic Structures (proper) (SSs), 7
Syntax of EL, 13
The movement interpretation rule (Büring), 107
The Perfect auxiliary, 65
The Progressive auxiliary be, 74
The specific de re reading, 36
The syntax of IL, 28

The temporal auxiliary be, 63
trace, 11
two-place distributor, 88
type clash, 18
type system for EL, 13
type system for IL, 27
Type-lifting the object position, 100
unselective binder, 111
Var, 14, 15, 29
Weak Crossover, 107
WH, 24
WHETHER, 40
whether-question, 40
WH-PRO, 39
will, 66
XN-Perfect, 66
yesterday, 66

19. REFERENCES

Baker, C. L. 1968. Indirect Questions in English, Linguistics Department, University of Illinois, Urbana.
Beck, Sigrid, and Sauerland, Uli. 2000. Cumulation is needed: A reply to Winter
Elbourne, Paul D. 2005. Situations and Individuals. Cambridge,


Fodor, J. A. 1970. The Linguistic Description of Opaque Contexts, Linguistics Department, MIT.


Heim, Irene. 1994a. Comments on Abusch's theory of tense: Manuscript, MIT.


Heim, Irene. 1994d. Puzzling reflexive pronouns in de se reports: Handout from Bielefeld conference.

Heim, Irene. 1994e. Questions: MIT, Lecture notes.


Academic Publishers.
zeitgenössischer Forschung*, eds. Arnim von Stechow and Dieter
Inferences*. Amsterdam: North-Holland.
*Approaches to Natural Language. Proceedings of the 1970 Stanford
Workshop on Grammar and Semantics*, eds. J. Hintikka, J. Moravcsik
Interactions with Temporal Adverbials*: Studies in Linguistics and
Ogihara, Toshiyuki. 1989. Temporal Reference in English and Japanese:
University of Texas at Austin.
In *Properties, Types and Meaning. Vol. 2*, eds. Gennaro Chierchia,
Penka, Doris. 2007. Negative Indefinites, Neuphilologische Fakultät der
Universität Tübingen/Seminar für Sprachwissenschaft, Universität
Tübingen: PhD Dissertation.
Pollock, Jean-Yves. 1989. Verb Movement, Universal Grammar and the
Cambridge, Mass.


