5. Predicates, Modifiers, Definite NPs, X-bar Theory, and Types

In this section we will extend the toy grammar developed so far. In particular, we will deal with definite and indefinite noun phrases, such as *a lemon* and *the lemon*, and with adjectives, such as *big lemon.*

5.1. Predicational Expressions

5.1.1. Predicational Adjectives

In this section we want to describe examples involving **adjectives** like the following:

(1) a. Molly is awake.
 b. Leopold is tired.
 c. Stephen is intelligent.

The adjectives *awake, tired* and *intelligent* describe certain properties of people. So we should assume that they have meanings of the following type:

(2) $[\![\textit{awake}]\!] = \lambda x \quad D_e$[x is awake][1]

That is, *awake* maps entities x to truth values, in particular, to 1 if x is awake, and else to 0. So they have a meaning similar to VPs, like *snores.* However, in order to form a sentence with a subject noun phrase, adjectives need a **copula**, like *is* (or *becomes* or *stays* or *remains*, which in addition denote certain temporal properties). What is the semantic contribution of *is*? Arguably, nothing at all, for our purposes at least: The adjective is already of the proper type to combine with an NP. Of course, the copula will carry tense information, such as the distinction between present, past and future. But otherwise it does not contribute much, and this explains why in many languages we do not need any copula, as e.g. in Russian.

How can we implement the idea that the copula does not contribute anything to the meaning of the adjective? We can do this by letting it denote the **identity function** for verb phrases. This is a function that takes a verb phrase meaning and gives back the same meaning:

(3) $[\![\textit{is}]\!] = \lambda P \quad D_{et}[P]$

Alternatively, we could have specified this function as $\lambda P \quad D_{et}\lambda x \quad D_e[P(x)]$.

Now all that's left is to add the necessary syntactic rules. I use "Cop" for copula, and "AP" for adjective phrase.

(4) a. VP → Cop AP
 b. AP → {*awake, tired, intelligent...*}
 c. Cop → {*is*}

We now can analyze our example sentence as follows:

[1] Perhaps we want to restrict adjectives like *awake* to people and animals. We could do so by giving the meaning λx[x is a person or an animal | x is awake]. In general, I will disregard such restrictions here.

(5) a. $[\![[_{S}\ [_{NP}\ \textit{Molly}]\ [_{VP}\ [_{Cop}\ \textit{is}]\ [_{AP}\ \textit{awake}]]]]\!]$
b. $= [\![[_{VP}\ [_{Cop}\ \textit{is}]\ [_{AP}\ \textit{awake}]]]\!]([\![[_{NP}\ \textit{Molly}]]\!])$
c. $= [\![\textit{is}]\!]([\![\textit{awake}]\!])([\![\textit{Molly}]\!])$
d. $= \lambda P \quad D_{et}[P](\lambda x \quad D_{e}[x \text{ is awake}])(MB)$
e. $= \lambda x \quad D_{e}[x \text{ is awake}](MB)$
f. = [MB is awake]
g. = 1, if MB is awake,
= 0, else.

5.1.2. Prepositional Phrases

Another type of postcopular expression are prepositional phrases (PP's) like the following:

(6) Molly is in Dublin.

Prepositional phrases consist of a preposition, here *in*, and an NP, here the name *Dublin*. We can assume the following syntactic rules:

(7) a. PP → P NP
b. P → {*in*, *on*, *at* …}
c. VP → Cop PP

The meaning of a PP, at least in postcopular position, should be similar to the meaning of an adjective. That is, it should denote something in the domain D_{et}. This means that the preposition itself takes an NP meaning (domain D_e) and yields some meaning within D_{et}. Example:

(8) $[\![\textit{in}]\!] = \lambda x \quad D_{e}\lambda y \quad D_{e}[y \text{ is in } x]$

We can analyze (6) as follows:

(9) a. $[\![[_{S}\ [_{NP}\ \textit{Molly}]\ [_{VP}\ [_{Cop}\ \textit{is}]\ [_{PP}\ [_{P}\ \textit{in}]\ [_{NP}\ \textit{Dublin}]]]]]\!]$
b. $= [\![[_{VP}\ [_{Cop}\ \textit{is}]\ [_{PP}\ [_{P}\ \textit{in}]\ [_{NP}\ \textit{Dublin}]]]]\!]([\![[_{NP}\ \textit{Molly}]]\!])$
c. $= [\![\textit{is}]\!]([\![\textit{in}]\!]([\![\textit{Dublin}]\!]))([\![\textit{Molly}]\!])$
d. $= \lambda P \quad D_{et}[P](\lambda x \quad D_{e}[\lambda y \quad D_{e}[y \text{ is in } x]])(\text{Dublin}))(MB)$
e. $= \lambda P \quad D_{et}[P](\lambda y \quad D_{e}[y \text{ is in Dublin}])(MB)$
f. $= \lambda y \quad D_{e}[y \text{ is in Dublin}](MB)$
g. = [MB is in Dublin]

5.2. Definite NPs

5.2.1. Russell's Analysis vs. Strawson's Analysis

Let us now turn to definite NPs, such as *the book* or *the woman.* What do such expressions mean? A plausible understanding is that, for example, *the woman* stands for Molly if Molly is the only woman around, that is, the only woman in the domain. A sentence with a definite NP then has a meaning as illustrated with the following example:

(10) *The woman snores* has the same truth value as *Molly snores*,
if Molly is a woman, if there is no other woman, and if Molly snores.

Or, more generally:

(11) *The woman snores* has the truth value 1 if the following conditions hold:

a. There is a woman,
b. there is not more than one woman,
c. and a woman snores.

Else, the truth value is 0.

This is, in essence, the analysis of definite NPs (or, “definite descriptions”) given in a famous article by Bertrand Russell, published in 1905 (“On denoting”). The condition (a) is called the **existence** condition, and the condition (b) is called the **uniqueness condition** of definite NPs.

A peculiar property of Russell’s analysis is that the expression *the woman* does not really mean anything in isolation. It rather contributes in an indirect way to the meaning of the whole sentence, but it does not stand for a particular person, like proper names.

Also, Russell’s analysis entails that the sentence is false if one of the three conditions is not met, that is, if there is no woman, if there are two or more women, or if the unique woman does not snore. This has been criticized by Peter Strawson in 1950, who pointed out that conditions (a) and (b) are of a different nature than condition (c). Taking up an example by Russell, Strawson argued that the following sentence is not simply false but rather quite inadequate when uttered nowadays, as France is a republic and has no king.

(12)The present king of France is bald.

In medieval times, when there were two, and sometimes three popes, the sentence *the pope is bald* could not be judged as either true or false, as it is unclear what the phrase *the pope* refers to.

Strawson argued that conditions (11.a) and (b) are **presuppositions** of the definite description *the woman*. As presuppositions they have to be satisfied; otherwise, the definite NP cannot even be interpreted. Consequently, the sentence in which the definite NP occurs cannot be interpreted, and therefore it cannot be judged as true or false.

Strawson’s treatment takes up the theory of definite descriptions suggested by Frege. Frege assumed that definite descriptions indeed refer, just like names, but only under the condition that existence and uniqueness are satisfied. For example, Frege (1892) analyzes the expression *the negative square root of 4*:

> We have here a case in which out of a concept-expression [that is, *negative square root of 4*] a compound proper name is formed with the help of the definite article in the singular, which is at any rate **permissible** if **one and only one** object falls under the concept.

Frege suggests here that the formation of proper names from a “concept-expression” is not permissible if no object falls under the concept, or if more than one object falls under the concept.

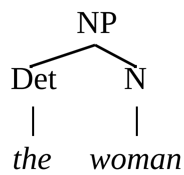
5.2.2. Definite NPs and Conditions on the Domain

Let us try to give an analysis of definite NPs like *the woman* in terms of the Frege/Strawson analysis. As for the syntax we can assume the following rules:

(13)a. NP → Det N
b. Det → *the*
c. N → {*woman*, *man*, *book*, …}

This will give us structures like the following:

(14)



What is the meaning of a noun? For the time being, a good choice is that a noun has the same type of meaning as an adjective. For example, the noun *woman* denotes a function that maps entities to 1 if they are a woman, and else to 0:

(15) ⟦*woman*⟧ = $\lambda x \;\; D_e$[x is a woman]

We can now express Frege's condition for definite NPs in our framework as follows. The definite article *the* denotes a function with a domain that comprises all those functions in D_{et} that map exactly one element of D_e to the truth value 1, and all others to 0. The value of *the* is then that element. For other functions in D_{et} the meaning of *the* is undefined. One way to express this definition is as follows:

(16) ⟦*the*⟧(P): Defined if there is one and only one x $\;\; D_e$ such that P(x) = 1.
If defined, ⟦*the*⟧(P) = x.

We can describe such functions with lambda terms, as follows:

(17) ⟦*the*⟧ = $\lambda P \;\; D_{et}$[#P = 1 | ιP]

Recall that the notation λX[…|---] is intended to express the restriction of the function in the "…"-part. Here #P stands for the number of elements in the predicate P, more precisely, for the number of elements that the characteristic function P maps to 1. It can be defined with the help of the cardinality function for sets, for which we also use the caret symbol, #:

(18) For P $\;\; D_{et}$: #P = #{x $\;\; D_e$ | P(x) = 1}

And ιP stands for the element that P applies to. Here, ι is the so-called iota operator. It is defined, for our purposes[2], as follows:

(19) For P $\;\; D_{et}$: then ιP = x iff P(x) = 1 and for all y such that P(y) = 1 it holds that x = y.

That is, ιP stands for an individual x if P(x) is true, and for all y such that P(y) is true, x and y are equal. This means that there is an individual to which P applies, and there is only one such individual. If these conditions do not hold, then ιP is undefined.

Assume that the universe D_e contains exactly one woman (say, Molly). Then the sentence *The woman snores* gets analyzed as follows:

[2] The standard definition includes a variable; we would then write ιx[P(x)].

(20)a. $[\![[_S [_{NP} [_{Det}\ \textit{the}] [_N\ \textit{woman}]] [_{VP} [_V\ \textit{snores}]]]]\!]$
b. $= [\![[_{VP} [_V\ \textit{snores}]]]\!]([\![[_{NP} [_{Det}\ \textit{the}] [_N\ \textit{woman}]]]\!])$
c. $= [\![\textit{snores}]\!]([\![\textit{the}]\!]([\![\textit{woman}]\!])$
d. $= \lambda x \quad D_e[\text{x snores}](\lambda P[\ \#P = 1 \mid \iota P](\lambda y[\text{y is a woman}]))$
e. $= \lambda x \quad D_e[\text{x snores}](\iota\lambda y[\text{y is a woman}])$,
provided that $\#\lambda x[\text{x is a woman}] = 1$, else undefined.
f. $= [\iota\lambda y[\text{y is a woman}]\ \text{snores}]$,
provided that $\#\lambda x[\text{x is a woman}] = 1$, else undefined.
g. $= 1$, if $\iota\lambda y[\text{y is a woman}]$ snores
$= 0$, if $\iota\lambda y[\text{y is a woman}]$ does not snore,
undefined, if $\#\lambda x[\text{x is a woman}] \neq 1$

Note that $\iota\lambda y[\text{y is a woman}]$ stands for the only woman in the domain. This is the expected result: If there is no woman, or more than one woman, the sentence is undefined. Otherwise, the sentence is true if the woman snores, and false, if she doesn't.

5.2.3. Definite NPs in Context

In many cases the uniqueness condition of an definite NP is not satisfied, and speakers still can use a definite NP without problem. One case in point is the following:

(21)The president is on a state visit in China.

Uttered in the US, (21) will typically refer to the current president of the US. This holds even though there are many presidents in the world — presidents of states and others —, and there are even many presidents of the US in history. In a sense, the definite NP picks out the most **salient** individual in the denotation of the noun *president*. We can handle this in one of two ways:

· Either we change the interpretation rule for the definite article, saying that an NP of the form *the N* picks out the **unique** object that falls under the predicate *N*, or that it picks out the most **salient** object that falls under the predicate *N*.

· Or we allow that the notion of a domain of discourse D_e may vary appropriately. When we utter *the president* in (21), then we can assume that the domain of discourse shrinks to a set C that includes no other president but the current president of the US. (C is the **contextually salient domain of discourse**).

The first approach is perhaps not so not nice because it claims that the definite article is inherently ambiguous. The second approach has empirical problems when it comes to sentences like the following:

(22)The president talked to the president of China.

To follow that approach would force us to a strange position, namely, that the contextually salient domain of discourse can widen during a single clause.

Presumably we should assume that definite NPs in general refer to **the most salient entity under the given description**. If the description applies to just one entity, then this will be the most salient one (there is no other, after all!) If the description applies to no entity at all, then the definite NP will be inadequate. If there is more than one entity, then the definite description will refer to the most salient one. Only if the two entities rank equally in saliency, the definite description will be inadequate.

The role of saliency can also be observed in the use of definite descriptions to refer to entities that play a role in a text:

(23) When Leopold entered the pub, a man was sitting in the corner. Leopold ordered a pint of guiness and opened the newspaper. After several minutes, the door opened and another man came in. Leopold didn't notice him. *The man* came up to him and put his hand on Leopolds' shoulder.

Notice that in this text, the definite NP *the man* most likely refers to the man that came in, not the man that was sitting in the corner. As the man that came in was mentioned more recently, he is presumably more salient.

5.3. Nominal Modifiers

5.3.1. Modifying Adjectives

So far we have considered nouns that are syntactically simple — nouns like *woman* and *apple.* But nouns can be modified by adjectives, as in *tired women* and *yellow apple*, and as such they can be part of an NP, as in *a tired woman* or *the yellow apple.* We can capture this with the following syntactic rule:

(24) N $\rightarrow$ AP N

This will generate syntactically complex nouns like (25.a), which can become parts of noun phrases as in (b).

(25) a. $[_{N}[_{AP}$ *yellow*$]$ $[_{N}$ *apple*$]]$
b. $[_{NP}$ $[_{Det}$ *the*$]$ $[_{N}[_{AP}$ *yellow*$]$ $[_{N}$ *apple*$]]]$

What about the semantics of an expression like (25.a), *yellow apple*? As it stands, semantic interpretation would crash: It is a complex expression that must be interpreted by functional application. But the meaning of *yellow* is in D_{et}, and the meaning of *apple* is in D_{et}, hence functional application is impossible.

What a complex noun like *yellow apple* should mean is relatively obvious: It should be a function that maps every entity to 1 if it is both yellow and an apple, and else to 0, or even to undefined, if the entity is not defined for either the meaning of *yellow* or the meaning of *apple.*

It is easy to define the particular semantic combination that we need here. With the help of the conjunction operator MIN we can give the meaning of *yellow apple* in terms of its immediate parts as follows:

(26) $[\![[_{NP} [_{AP} \textit{yellow}] [_{N} \textit{apple}]]]\!] = \lambda x \quad D_e[\text{MIN}\{[\![[_{AP} \textit{yellow}]]\!](x)\ [\![[_{N} \textit{apple}]]\!](x)\}]$

This is the function that maps an entity x to 1 if both the meanings of *yellow* and of *apple* map it to 1, and else to 0. Let us perhaps simplify this a bit and use from now on the symbol for conjunction in propositional logic   instead of the MIN-function (and similarly   and ¬).

(27) $[\![[_{NP} [_{AP} \textit{yellow}] [_{N} \textit{apple}]]]\!] = \lambda x \quad D_e[[\![[_{AP} \textit{yellow}]]\!](x) \quad [\![[_{N} \textit{apple}]]\!](x)]$

We can express this in the following interpretation rule of **Intersective Combination**:

(28) **Intersective Combination:**
$[\![[\alpha\ \beta]]\!] = \lambda x \quad [\text{DOM}([\![\alpha]\!]) \cap \text{DOM}([\![\beta]\!])]\ [\ [\![\alpha]\!](x) \quad [\![\beta]\!](x)]$, if $[\![\alpha]\!]$, $[\![\beta]\!]$ $\quad D_{et}$.

That is, the meaning of [α, β], where the meanings of both belong to D_{et}, is a function from entities to truth values as well. This function maps every entity x to 1 if both the meaning of α and the meaning of β maps it to 1, and else to 0. The function is defined for x if and only if x is both defined for the meaning of α and for the meaning of β.

This is one way of dealing with adjectives and also with prepositional phrases that modfy a noun. Another way is to actually change the meaning of APs and PPs for those cases slightly and in a systematic way, and stay with functional application as the only interpretation rule. The following meaning is the appropriate one for the attributive version of *yellow*, that is, if *yellow* occurs in pre-nominal position:

(29) $[\![\textit{yellow}]\!] = \lambda P\ \ D_{et}\lambda x\ \ D_e[P(x)\ \ \ [x \text{ is yellow}])]$

That is, *yellow* is a function that maps a function P in D_{et} (that is, P is a function from entities to truth values) to a function from entities to truth values. We then have the following derivation:

(30)a. $[\![[_N[_{AP}\ \textit{yellow}]\ [_N\ \textit{apple}]]]\!]$
b. $= [\![\textit{yellow}]\!]([\![\textit{apple}]\!])$
c. $= \lambda P\ \ D_{et}\lambda x\ \ D_e[P(x)\ \ \ [x \text{ is yellow}])](\lambda x\ \ D_e[x \text{ is an apple}])$
d. $= \lambda x\ \ D_e[[x \text{ is an apple}]\ \ \ [x \text{ is yellow}])]$

The question arises whether we should assume that *yellow* is ambiguous between a predicative use and an attributive use, or whether there are two distinct words that just happen to be pronounced the same. If we just look at English, the first version is clearly supported. We can assume that this is another instance of type flexibility: Adjective meanings happen to come in two types, called **predicative** and **attributive**. Predicative adjectives are functions from D_e to D_t, and attributive adjectives are functions from D_{et} to such functions. The two meanings are systematically related, in the following way:

(31)If α is the meaning of a predicative adjective,
then $\alpha' = \lambda P\ \ D_{et}\lambda x\ \ D_e[P(x)\ \ \ \alpha(x)]$ is the meaning of the corresponding attributive adjective.

In languages which show more agreement than English we often find morphological differences between predicative and attributive adjectives. Consider German:

(32)a. Der Apfel ist gelb. ‘The apple is yellow.’
b. Der gelbe Apfel liegt auf dem Tisch. ‘The yellow apple is on the table.’

Only attributive adjectives agree with their noun, in features like gender, number and case (cf. *gelbe* vs. *gelb*). We can assume that it is a semantic function of the agreement morpheme to change the adjective meaning from its predicative version to its attributive version. English shows some evidence for a difference between predicative and attributive adjectives as well; adjectives like *awake* and *asleep* can only be used in the attributive way.

Notice that the syntactic rule (24) is that it is **recursive**. That is, the rule N → AP N can be applied to the N on the right-hand side again, which gives us complex nouns like the following:

(33)a. $[_N\ [_{AP}\ \textit{yellow}]\ [_N\ \textit{apple}]]$
b. $[_N\ [_{AP}\ \textit{sweet}]\ [_N\ [_{AP}\ \textit{yellow}]\ [_N\ \textit{apple}]]]$
c. $[_N\ [_{AP}\ \textit{juicy}]\ [_N\ [_{AP}\ \textit{sweet}]\ [_N\ [_{AP}\ \textit{yellow}]\ [_N\ \textit{apple}]]]]$

As the output of applying an adjective to a noun results in a noun meaning again, there is no problem in dealing with such complex structures.

5.3.2. More on Adjective Meanings

There is much to be said about the variety of meanings that we find with adjectives. Here I will discuss some important structural differences in adjective meanings.

We have interpreted the adjective *yellow* in its attributive interpretation as restricting the entities that fall under the noun meaning to those that also fall under the meaning of the adjective, interpreted predicatively:

(34) $[\![[_N [_{AP}\ \textit{yellow}]\ [_N\ \textit{apple}]]]\!]$
$= \lambda x \quad D_e[\ [x \text{ is an apple}] \quad [x \text{ is yellow}])]$

This function maps all entities x to 1 that satisfy the following condition: x is an apple, and x is yellow. If we see the meaning of *yellow* and *apple* as sets (the set of yellow things, and the set of apples), then the meaning of *yellow apple* is the intersection of these sets. Therefore such adjectives are called **intersective**.

Not all adjectives are intersective. Consider the following examples:

(35) a. Bob is a tall boy.
b. Jim is a tall man.

To count as a tall boy one doesn't have to be as tall as a tall man (let alone as tall as a tall basketball player...). This is a problem for the intersective analysis of *tall* in (36.a), which tells us that a tall boy is something that is a boy and is tall, and that a tall man is something that is a man and is tall. A more adequate analysis is given in (b).

(36) a. Intersective analysis:
$[\![\textit{tall}]\!] = \lambda P \quad D_{et}[\lambda x \quad D_e[P(x) \quad [x \text{ is tall}])]]$
b. Non-intersective analysis:
$[\![\textit{tall}]\!] = \lambda P \quad D_{et}[\lambda x \quad D_e[$the size of x is greater than the average size of the elements in $\{x \mid P(x)\}]]$

According to (b), a tall boy is a boy whose size is greater than the average size of the boys. As the average size of the men may be higher, a tall boy need not be tall for a man.

We have considered two possible ways to deal with modifying adjectives. One was to introduce a new interpretation rule, essentially conjunction (cf. (28)); the other was to change the meaning of the adjective (cf. (29)). It appears that the first strategy can only deal with the intersective analysis, and so we should prefer the second strategy.

However, there is a problem in this argument. We find that the non-intersective analysis is not confined to attributive adjectives. We also find it also for predicative adjectives:

(37) a. Bob, who is still a boy, is tall.
b. Jim, who is a grown-up man by now, is tall.

As before, we would not like to interpret *tall* as identifying the same person in both sentences. But it is difficult to see how the solution that we proposed for attributive adjectives can be applied here, as the adjective does not have any noun meaning from which a non-intersective analysis could be derived.

Things become even more problematic if we consider the following cases. Here we have an attributive adjective that is applied to the same noun, but what counts as tall still is clearly different:

(38) a. The children in the kindergarten built a tall snowman.
b. The students in the senior high school built a tall snowman.

We expect snowmen built by students in a senior high school to be considerably taller than snowmen built by kindergarten children. But the meaning of *snowman* is the same in both cases, it applies to snowmen, regardless of the builder.

Clearly, the standards for tallness must be judged with respect to the whole sentence here, and there are certainly cases where even this is not enough and we even need information pertaining to the context in which the sentence is uttered. All we can do at this point is to indicate this context-dependency in a lexical rule:

(39) $[\![\textit{tall}]\!] = \lambda x \quad D_{et}[$x's size is greater than the size made salient by the utterance context$]$

But of course this is not a final analysis; we now would have to investigate the way how the utterance context makes salient a certain size.

Intersective adjectives like *yellow* and non-intersective adjectives like *tall* share one property, namely, that they restrict the meaning of the noun. A yellow apple is an apple; a tall boy is a boy. Therefore they are called **restrictive**. There are also **non-restrictive** adjectives:

(40) a. former senator
b. alleged thief
c. fake one-hundred dollar bill

A former senator is not a senator anymore, an alleged thief might not be a thief after all, and a fake one-hundred dollar bill is not really a one-hundred dollar bill.

Non-restrictive adjectives typically require us to imagine different circumstances, like previous times, circumstances that could be true but perhaps are not, and circumstances that a forger tried to present as true but which are not. In our current semantic framework we have no place to deal with such cases; they require the notion of possible worlds (cf. the introduction).

Many non-restrictive adjectives do not occur as predicative adjectives (cf. examples (25.a,b), but contrast with (c)):

(41) a. *This senator is former.
b. *This thief is alleged.
c. This one-hundred dollar bill is fake.

This suggests that there are at least some adjectives that genuinely have to be analyzed as modifying a noun. That is, not every adjective can be derived from simple predicative adjectives.

5.3.3. Modifying Prepositional Phrases

Prepositional phrases can be used as noun modifiers as well:

(42) a. $[_{N}$ $[_{N}$ *woman*$]$ $[_{PP}$ $[_{P}$ *in*$]$ $[_{NP}$ *Dublin*$]]]$

b. $[_{N}$ $[_{N}$ *apple*] $[_{PP}$ $[_{P}$ *on*] $[_{NP}$ $[_{Det}$ *the*] $[_{N}$ *table*]]]]]

Just as we distinguished between predicative uses of adjectives and attributive uses, we can distinguish between predicative and attributive uses of prepositional phrases. First, we have to assume the following syntactic rule:

(43) N $\rightarrow$ N PP

In contrast to adjectives, which typically precede their noun, prepositional phrases follow their noun in English.

We have to assign appropriate interpretation rules for PP's. Just as for adjectives, we have two options:

- First, we can let the interpretation rule for intersective combinations (28) handle things (notice that both noun meanings and PP meanings are in D_{et}).

(44) ⟦$[_{N}$ $[_{N}$ *apple*] $[_{PP}$ $[_{P}$ *on*] $[_{NP}$ $[_{Det}$ *the*] $[_{N}$ *table*]]]]]⟧
= λx D_e[MIN{⟦$[_{N}$ *apple*]⟧(x), ⟦$[_{PP}$ $[_{P}$ *on*] $[_{NP}$ $[_{Det}$ *the*] $[_{N}$ *table*]]]]⟧(x)}]

- Second, we can assume that PP meanings belong not only to D_{et}, the functions from D_e to D_t, but also functions from D_{et} into such functions:

(45) a. ⟦*in Dublin*⟧, in predicative meaning: λx D_e[x is in Dublin]
b. ⟦*in Dublin*⟧, attributive meaning: λP D_{et}[λx D_e[MIN({P(x), [x is in Dublin]})]]

The second meaning can either be derived by allowing general type flexibility for PP's, following a rule similar to (31), or by allowing a similar type flexibility already for the meaning of the preposition, here *in.*

The latter meaning is also suitable for PPs in adverbial function, as in the following example:

(46) ⟦*sleep in Dublin*⟧
= ⟦*in Dublin*⟧(⟦*sleep*⟧)
= λP D_{et}[λx D_e[MIN({P(x), [x is in Dublin]})]](λx D_e[x sleeps])
= λx D_e[D_e[MIN([x sleeps], [x is in Dublin]})]

5.3.4. Definite NPs and Noun Modification

In our rule for attributive PP's we have suggested that they are attached to nouns, and not to noun phrases. That is, we analyze the phrase *the fly on Molly* as in (47.a), and not as in (b).

(47) a. $[_{NP}$ $[_{Det}$ *the*] $[_{N}$ $[_{N}$ *fly*] $[_{PP}$ *on Molly*]]]

b. $[_{NP}$ $[_{NP}$ $[_{Det}$ *the*] $[_{N}$ *fly*]] $[_{PP}$ *on Molly*]]

For purely syntactic reasons we could have assumed the structure (b), with a corresponding syntactic rule NP $\rightarrow$ NP PP. One reason for that might be that it corresponds better to the prosodic structure of this expression, which is quite clearly (*the fly*)(*on Molly*), not (*the*)(*fly on Molly*) or (*the*)(*fly*)(*on Molly*).

However, for semantic reasons the structure (47.a) is more plausible. Consider a situation in which we have two flies, and one fly sits on Molly. Clearly, the phrase *the fly on Molly* is defined in this situation, and refers to the apple on the plate. This is what our grammar indeed tells us if we analyze the phrase according to (47.a):

(48) a. ⟦$[_{NP}$ $[_{Det}$ *the*] $[_{N}$ $[_{N}$ *fly*] $[_{PP}$ *on Molly*]]]⟧

b. = ⟦*the*⟧(⟦*on Molly*⟧(⟦*fly*⟧))

c. = ⟦*the*⟧(($\lambda P \quad D_{et}\lambda y \quad D_e$[MIN({P(y), y is on Molly})]($\lambda z \quad D_e$[z is a fly]))

d. = $\lambda P \quad$ {Q $\quad D_{et}$ | #{x $\quad D_e$ | P(x)} = 1}[ιP]($\lambda y \quad D_e$[MIN({[z is a fly], [z is on Molly]})])

e. = ι[$\lambda y \quad D_e$[MIN({[z is a fly], [z is on Molly]})]],
if #{x $\quad D_e$ | #{x $\quad D_e$ | MIN({[x is a fly], [x is on Molly]})]} = 1},
else undefined.

That is, the expression is defined only if there is exactly one x such that x is a fly and x is on Molly. This is the case in our example. In the alternative analysis,

(49) a. ⟦$[_{NP}$ $[_{NP}$ $[_{Det}$ *the*] $[_{N}$ *fly*]] $[_{PP}$ *on Molly*]]⟧

b. = ⟦*on Molly*⟧(⟦$[_{NP}$ $[_{Det}$ *the*] $[_{N}$ *fly*]]⟧)

we would have to compute the meaning of the expression *the fly*. But as we assumed that there are two flies, the meaning of this expression is not defined.

5.4. Arguments, Adjuncts and X-bar Theory

5.4.1. Argument and Adjuncts

In this chapter we have discussed two quite different ways to derive a complex phrase. Consider the following examples:

(50) a. Molly sang the aria.
b. Molly sang in Dublin.

In (a), *sang* is a transitive verb that is combined with an object NP. In (b), *sang* is an intransitive verb that is combined with a prepositional phrase. The object NP satisfies the object argument position of the verb. The PP does not satisfy any argument position; the result is of the same type, D_{et} (In the following derivation, I distingish between the transitive *sing* and the intranstive *sing′*).

(51) a. ⟦$[_{VP}$ $[_{V}$*sang*] $[_{NP}$ *the aria*]]⟧

b. = ⟦*sang*⟧(⟦*the aria*⟧)

c. = $\lambda x \quad D_e\lambda y \quad D_e$[y sang x](a)

d. = $\lambda y \quad D_e$[y sang a]

(52) a. ⟦$[_{VP}$ $[_{VP}$ $[_{V}$ *sang′*]] $[_{PP}$ *in Dublin*]]⟧

b. = ⟦*in Dublin*⟧(⟦*sang′*⟧)

c. = $\lambda P \quad D_{et}\lambda x \quad D_e$[MIN{P(x), [x is in Dublin]}]($\lambda x \quad D_e$[x sang])

d. = $\lambda x \quad D_e$[MIN{[x sang], [x is in Dublin]}]

The NP *the aria* is called an **argument**, and the PP *in Dublin* an **adjunct** in those constructions. The other subexpression is called the **head**. There are a number of differences between arguments and adjuncts. Most importantly, an argument can occur only once; as soon as the argument slot is filled, no other argument of the same type can occur. But there can be more than one adjunct.

(53)a. *Molly sang the aria the song.

b. Molly sang in Dublin in the evening at the beach.

This follows from the fact that an adjunct simply modifies its head, returning some expression with a similar (but perhaps more restricted) meaning.

Our syntactic rules mirror this behavior to a certain extent. Argument-head constructions typically have the form (54.a), with X≠Y≠Z, whereas adjunct-head constructions have the form (b), in which the head has the same category label as the whole construction (here, X).

(54)a. $[_X [_Y \alpha] [_Z \beta]]$

b. $[_X [_X \alpha] [_Y \beta]]$ (or $[_X [_Y \beta] [_X \alpha]]$)

In American Structuralism, head-argument constructions were called **exocentric** because the “center” (the head) was of a different category than the whole, and head-adjunct constructions where called **endocentric** because the head was of the same category.

5.4.2. Other Head-Argument and Head-Adjunct Constructions

There are quite a few other instances of the two construction types we just have mentioned. For example, adverbials like *loudly* are adjuncts:

(55)Leopold $[_{VP} [_{VP}$ snored$] [_{AdvP}$ loudly$]]$.

We have to assume syntactic rules like VP → VP AdvP, and interpretations as the following (which are specified in a rather preliminary way here).

(56) ⟦*loudly*⟧ = $\lambda P \;\; D_{et} \lambda x \;\; D_e$[P(x) and x does P in a loud way]

We have seen that prepositional phrases can be used in the same way:

(57)Molly $[_{VP} [_{VP} [_V$ sings$]\ ] [_{PP}$ in Dublin$]]$

But prepositional phrases are not always modifiers. Take the case *Molly resides in Dublin*; this certainly does not mean that Molly resides and that she is in Dublin. The sentence *Molly resides* does not make much sense in the first place, and to say of someone that she resides in Dublin does not mean that she has to be in Dublin at this very moment. Verbs like *reside* arguably take a PP as an argument. At the current state of our discussion we can perhaps assign them meanings of the following kind:

(58) ⟦*reside*⟧ = $\lambda P \;\; D_{et} \lambda x \;\; D_e$[there is a y, x resides at y, and P(y)]

We then get analyses like the following, where I have assumed the predicative meaning for the prepositional phrase:

(59)a. ⟦$[_S [_{NP}$ *Molly*$] [_{VP} [_V$ *resides*$] [_{PP}$ *in Dublin*$]]]$⟧

b. = ⟦*resides*⟧(⟦*in Dublin*⟧)(⟦*Molly*⟧)

c. $= \lambda P \quad D_{et} \lambda x \quad D_e$[there is a y, x resides at y, and P(y)]($\lambda z \quad D_e$[z is in Dublin])(MB)

d. = [there is a y, MB resides at y, and y is in Dublin]

As I indicated, this analysis is still fairly preliminary — for example, we get a pretty redundant translation for *Molly resides at Eccles street 9*. What appears important is that *in*, in all examples considered so far, expresses a spatial relationship. All PPs expressing spatial relationships can occur with *reside*, e.g. *reside under the bridge*, *reside over the valley*, *reside behind the church* etc.

But I should add here right away that *in* does not always denote a spatial relationship. Cases like the following are rather idiosyncratic cases in which the verb requires for some arcane grammatical purpose an argument marked by *in*:

(60)Molly believes in God.

Let me mention one other head-argument construction. Many nouns do not simply express a property (a function from entities to truth values), but come with an argument position. This is typically filled by a PP with the preposition *of*.

(61)a. the [father [$_{PP}$ of Molly]]

b. a [sister [$_{PP}$ of Molly]]

Such nouns are called **relational**, or, in the case the relation is a function, **functional**. For example, the meaning of *sister* can be given as follows:

(62) ⟦*sister*⟧ = $\lambda x \quad D_e \lambda y \quad D_e$[y is a sister of x]

Which constituent should we assign to relational nouns? If we just assume that they are of category N, we cannot distinguish between cases in which the PP is a modifier, and cases in which it is an argument. One way to go that has been suggested is the following. We reserve the term "NP" for what we have called "N" so far, that is, for nominal expressions that are functions from entitities (D_e) to truth values (D_t). A relational noun like *sister* is of category N; when we combine it with an *of*-phrase, we get an expression of the category NP. What we have called "NP" so far we call **determiner phrase**, DP. It typically consists of a determiner, Det, and an NP. We then have rules like the following:

(63)a. NP $\rightarrow$ N PP_{of}

b. $PP_{of} \rightarrow P_{of}$ DP, $\quad P_{of} \rightarrow$ *of*

c. DP $\rightarrow$ Det NP

d. DP $\rightarrow$ *Molly, Leopold, Stephen*

Other rules have to be changed accordingly, to reflect the change from NP to DP. For example, we now have rules like NP $\rightarrow$ AP NP, NP $\rightarrow$ NP PP, or S $\rightarrow$ DP VP.

With rule (63.a) it is clear that the *of*-PP fills an argument position of the noun N. The preposition *of* is semantically empty (the identity function); it takes an entity and gives back the same entity.

(64) ⟦*of*⟧ = $\lambda x \quad D_e$[x]

We have derivations like the following:

(65) a. ⟦[$_{NP}$ [$_{N}$ *sister*] [$_{PPof}$ [$_{Pof}$ *of*] [$_{DP}$ *Molly*]]]⟧

b. = ⟦*sister*⟧(⟦*of*⟧(⟦*Molly*⟧))

c. = $\lambda x \ \ D_e \lambda \ \ D_e$[y is a sister of x]($\lambda x \ \ D_e$[x](MB))

d. = $\lambda x \ \ D_e \lambda y \ \ D_e$[y is a sister of x](MB)

e. = $\lambda y \ \ D_e$[y is a sister of MB]

This is different from cases in which the PP is a modifier, which are essentially analyzed as before:

(66) a. ⟦[$_{NP}$ [$_{NP}$*woman*] [$_{PP}$ [$_{P}$ *in*] [$_{DP}$ *Dublin*]]]⟧

b. = ⟦*in*⟧(⟦*Dublin*⟧)(⟦*woman*⟧)

c. = $\lambda y \ \ D_e \lambda P \ \ D_{et} \lambda x \ \ D_e$[MIN({P(x), [x is in y]})](Dublin)($\lambda z \ \ D_e$[z is a woman])

d. = $\lambda x \ \ D_e$[MIN({[x is a woman], [x is in Dublin]})]

5.4.3. X-bar Theory

One prominent framework of syntactic description, called **X-bar theory** (Chomsky 1970, Jackendoff 1977) tries to capture the nature of the head-argument construction and the head-adjunct construction across different categories by a special system of naming the categories. It distinguishes between different levels of syntactic categories by bars, primes, or indices. For example, X^0 is a basic or **lexical** constituent of the type X, and X′ is a level with one argument position saturated, and XP is a fully saturated constituent, a so-called **maximal constituent**. The two construction types then can be characterized as follows:

(67) a. Head-adjunct constructions: $X^n \rightarrow$ YP X^n (or X^n YP)
b. Head-argument constructions: $X^{n+1} \rightarrow$ YP X^n (or X^n YP)

That is, in a head-adjunct construction the modifier YP does not change the bar-level of the head category, whereas in a head-argument construction the argument YP increases the bar-level by one. Also, the adjunct and the argument are maximal constituents (YP). Different versions of X-bar theory differ in various respects from this setup, but for our current purposes this characterization should be sufficient.

When we try to implement this naming convention into our syntax of Toy English, we find that there is one rule that somehow doesn’t fit very well. This is the basic rule S $\rightarrow$ DP VP. For one thing, the subject DP is an argument of the VP, for semantic reasons. But this does not show up in the rule: For one thing, the mother node, S, and the head daughter node, VP, are named differently, and secondly, the head daughter is a maximal constituent and hence should not have any arguments positions that still need to be filled. So it has been proposed to use the term “VP” essentially for what we have called “S”, and assume that the subject is the outermost argument of the VP. We then have rules like (68.a) for the subject argument and the object argument, and constructions like (b).

(68) a. VP $\rightarrow$ DP V′,   V′ $\rightarrow$ V^0 (DP)

b. $[_{VP}\ [_{DP}$ *Leopold*$]\ [_{V'}\ [_{V}$ *loves*$]\ [_{DP}$ *Molly*$]]]$

There is one rule we haven't mentioned so far, namely, the rule VP $\rightarrow$ Aux VP_{inf} that we employed in the last chapter for negated sentences, like *Molly doesn't snore*. We have to employ a rule like that for a variety of other sentence types as well:

(69) a. Molly does snore.
b. Molly will snore.
c. Molly might snore.
d. Molly has snored.
e. Molly is snoring.

In (d), the verb of the VP is not in the regular infinite form, but rather in a **past participle** form, and in (e) it is a **present participle**. These are varieties of infinitives that we will simply neglect here. Also, it is tempting to see constructions without auxiliary as cases in which the contribution of the auxiliary, which is expressed by agreement with the subject and information such as tense, is incorporated in the verb itself, perhaps residing in the inflectional suffix:

(70) a. Molly snore**s**.
b. Molly snore**d**.

The rule VP $\rightarrow$ Aux VP_{inf} that we have assumed so far may look like an adjunct-head construction with VP as head, as the category VP expands into something that contains a similar category, VP_{inf}. But notice that there is a category change after all, from VP_{inf} to VP. It has been argued that the auxiliary is the head of this construction, and as it carries the inflection, it is often called the "IP", short for **inflection phrase**.

We have now proposed two distinct changes that affect our previous rule S $\rightarrow$ NP VP: First, we have argued that S should be called "VP" all along, and second, we suggested that we have a separate phrase level over VP, namely, IP, that captures auxiliaries and perhaps also the inflection of finite main verbs. How can these two changes be implemented together? A first try is the following:

(71) a. IP $\rightarrow$ I^0 VP

b. VP $\rightarrow$ DP V′,   V′ $\rightarrow$ V^0 DP

But this would give us the wrong word order. We would get structures like the following one, which are fine for questions but not good for declaratives.

(72) $[_{IP}\ [_{I0}$ *will*$]\ [_{VP}\ [_{DP}$ *Molly*$]\ [_{V'}\ [_{V0}$ *love*$]\ [_{DP}$ *Leopold*$]]]]$

Evidently, the subject must precede the auxiliary. We should assume that the rule that expands IP provides for a syntactic position in which the subject can be realized:

(73) a. IP $\rightarrow$ DP I′,

b. I′ $\rightarrow$ I^0 VP

Now this leads to another problem: Now we have two positions for the subject, one inside VP, and one outside VP but inside the IP. But subjects certainly do not ocur twice! One assumption that is often made is that it is filled at this position by an **empty** expression “e”, one that does not have any phonetic material but for which it is guaranteed that it is interpreted by the constituent that fills the position of DP under IP. We can assume that empty expressions are semantically inert, and that semantic interpretation makes sure that the argument position ultimately gets filled by the right expression. This is done in the following setup:

(74)a. $[\![e]\!] = \lambda P \quad D_{et}[P]$

b. $[\![\textit{will}]\!] = \lambda P \quad D_{et}\lambda x \quad D_{e}[\text{it will be the case that } P(x)]$

That is, the auxiliary *will* states that the sentence P(x) should hold in the future. (This is a very prelimary analysis of tenses of course.)

(75)a. $[\![[_{IP}\ [_{DP}\ \textit{Molly}]\ [_{I'}\ [_{I0}\ \textit{will}]\ [_{VP}\ [_{DP}\ e]\ [_{V'}\ [_{V0}\ \textit{snore}]]]]]]\!]$

b. $= [\![\textit{will}]\!]([\![e]\!]([\![\textit{snore}]\!]))([\![\textit{Molly}]\!])$

c. $= [\![\textit{will}]\!](\lambda P \quad D_{et}[P](\lambda x \quad D_{e}[x \text{ snores}]))(\text{MB})$

d. $= [\![\textit{will}]\!](\lambda x \quad D_{e}[x \text{ snores}])(\text{MB})$

e. $= \lambda P \quad D_{et}\lambda x \quad D_{e}[\text{it will be the case that } P(x)](\lambda x \quad D_{e}[x \text{ snores}])(\text{MB})$

f. $= [\text{it will be the case that } [\text{MB snores}]]$

This gives us the right interpretation. Notice that the empty subject DP does not carry out any work, but leaves the subject argument of the VP untouched; the meaning of the auxiliary *will* is such that it will later be filled by its argument. Also, the meaning of the subject NP ends up in the righ argument slot of the verb.

There are other ways to inform the subject argument of the VP about the main argument of the IP. We can endow our grammar with the possibility that constituents undergo **movement** from one position to another one, an option that appears very plausible when we consider cases like the following were clearly some kind of displacement of constituents has happened. I indicate the relation between moved the moved constituent and its original position by coindexation.

(76)a. Who$_1$ does Leopold love e$_1$?
b. Molly$_1$, Leopold loves e$_1$.

We can now assume that a similar movement is responsible for the fact that the subject argument of the VP shows up as the argument of the IP:

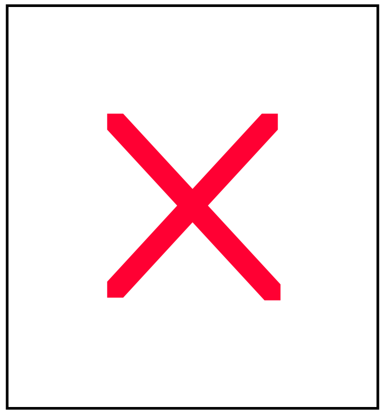
(77) $[_{IP}\ [_{DP}\ \text{Molly}]_1\ [_{I'}\ [_{I0}\ \text{will}]\ [_{VP}\ [_{DP}\ e]_1\ [_{V'}\ [_{V}\ \text{snore}]]]]]$

We of course would have to guarantee somehow that the moved constituent is interpreted in its original position. But this is an issue we will not pursue further here.

What about cases in which there is no overt auxiliary? It has been proposed that in this case the main verb moves into the slot provided by I^0, where it is combined with the tense information present in the inflectional suffix. Consider the following derivation of *Molly loves Leopold*:

(78)a. Before movement: $[_{IP}\ [_{DP}\ \text{Molly}]_1\ [_{I'}\ [_{I0}\ \text{-d}]_2\ [_{VP}\ [_{DP}\ e]_1\ [_{V'}\ [_{V0}\ \text{love}]_2\ [_{DP}\ \text{Leopold}]]]]]$

b. After movement: $[_{IP}\ [_{DP}$ Molly$]_1\ [_{I'}\ [_{I0}$ love-d$]_2\ [_{VP}\ [_{DP}$ e$]_1\ [_{V'}\ [_{V0}$ e$]_2\ [_{DP}$ Leopold$]]]]]$



Again we would have to make sure that the verb meaning gets interpreted in its original position.

Let us finally just list the changes that the X-bar rule format would bring about:

(79)a. IP → DP I′, I′ → I^0 VP

b. VP → DP V′ V′ → V^0 (DP)

c. DP → D^0 NP

e. PP → P^0 DP

f. NP → N^0 PP_{of}

g. NP → AP NP, NP → NP PP

h. VP → VP AdvP, VP → VP PP

5.5. Formal Definition of Types

The semantic objects we work with come in particular types. There are two basic, or **saturated** types, namely entities and truth values. We have called the set of entities D_e, the domain of discourse, and the set of truth values D_t (which we assumed to be the set {0, 1}).

We can built complex types out of these two types, which are **unsaturated**. For example, one frequently used type is the one of functions from entities to truth values. This is the type, for example, of the meaning of *woman*, which we have rendered as $\lambda x\ \ D_{et}$[x is a woman]. The set of objects belonging to this type we have called D_{et}.

It would come handy to have a general definition of all possible types, and a good way of naming these types. This can be done as follows:

(80)a. Basic types:
The basic types are e and t.

b. Complex types:
If σ, τ are types, then (σ)τ is a type.

We take (σ)τ to be the type of semantic objects that are functions from the domain of type σ to the domain of type τ. This can be expressed as follows:

(81)a. Domains of basic types:
D_e is the universe of discourse, D_t is the set of truth values {0, 1}

b. Domains of complex types:
If (σ)τ is a type, then $D_{(σ)τ}$ is the set of functions from $D_σ$ to $D_τ$.

For example, the type of functions from entities (type e) to truth values (type t) is (e)t, and the domain of this type is $D_{(e)t}$. We used to write D_{et} for that, and we will keep doing so because we adhere to the following abbreviation rule that allows us to drop parentheses around basic types.

(82)If (σ)τ is a type, and if σ stands for a basic type, then (σ)τ can be simply written as στ.

We can express other types as well with this system. Here are a few examples.

Type	Domain	Examples
et	D_{et}, functions from D_e to D_t	⟦*snore*⟧, = λx D_e[x snores] ⟦*yellow*⟧ (predicative), = λx D_e[x is yellow] ⟦*woman*⟧, = λx D_e[x is a woman]
eet	D_{eet}, functions from D_e to D_{et}	⟦*love*⟧, = λx D_eλy D_e[y loves x]
eeet	D_{eeet}, functions from D_e to D_{eet}	⟦*give*⟧ = λx D_eλy D_eλz D_e[z gives y to x]
tt	D_{tt}, functions from D_t to D_t	⟦*it is not the case that*⟧ = λt D_t[1—t]
ttt	D_{ttt}, functions from D_t to D_{tt}	⟦*and*⟧ for sentences, = λt D_tλt′ D_t[MIN({t, t′})]
(et)et	$D_{(et)et}$, functions from D_{et} to D_{et}	⟦*yellow*⟧, = λP D_{et}λx D_e[MIN({P(x), [x is yellow]})] ⟦*is*⟧, = λP D_{et}[P] ⟦*didn't*⟧ = λP D_{et}λx D_e[1 — P(x)]
e(et)et	$D_{e(et)et}$, f. from D_e to $D_{(et)et}$	⟦*in*⟧, = λx D_eλP D_{et}λy D_e[MIN({P(y), [y is in x]})]
ee	D_{ee}, function from D_e to D_e	⟦*of*⟧, = λx D_e[x]
(et)e	$D_{(et)e}$, functions from D_{et} to D_e	⟦*the*⟧, = λP[#{x D_e | P(x)} = 1 | ιP]

The last example shows that we have to understand the definition in (81.b) as saying that the $D_{(σ)τ}$ is the set of (potentially) **partial** functions from $D_σ$ to $D_τ$.

Let us come back for a moment to the general structure of the definitions (80) and (81). These are so-called **recursive definitions**. They define a set (a set of types, or a set of domains) in the following way: First, they give one or more basic cases. In (80) it is stated that e and t are types. Then they give one or more construction principles that allow us to derive new cases. In (80) we have one construction principle that says that if σ and τ are types, then (σ)τ is a type. Notice that we immediately have defined an infinite number of types. That is, there is no "biggest" type.

You should be aware that the classical way to define types uses the pair notation. Instead of “et”, we write “ e, t ”, instead of “eet” we write “ e, e, t ”, and instead of “(et)e” we write “ e, t , e ”. The notation adopted here is equivalent, but a bit shorter and often easier to read, especially when the types get more complex.