Mathematical Structures in Language

Edward L. Keenan **Lawrence S. Moss**

December 9, 2009

**CENTER FOR THE STUDY
OF LANGUAGE
AND INFORMATION**

Contents

Preface

The purpose of this book is twofold: First, we present elementary mathematical background to help the student of linguistics formulate generalizations concerning the structure of languages, most particularly syntactic and semantic structure. This is a goal we fully share with Partee et al. (1990), a book with similar goals to our own. And to stress a basic point: the background mathematical work we present is well known and well understood. The primary loci of this material is in Chapters 2, 3, 6, 7 and 9. But the linguistic phenomena we study are much less well understood—they are what we are trying to understand. A major step in understanding is formulating what we are trying to understand in terms of things we already understand, hence the drive towards mathematical formulation. This enables us to notice and formulate linguistic regularities in a clear and precise way, enabling us to study, correct, test and generalize them. So this is the core answer to a question we're frequently asked "Why study language mathematically?".

Our second purpose follows upon the first but is considerably more speculative. Namely, once our mathematical description of linguistic phenomena is sufficiently developed, we can not only model properties of natural language, we can study our models to derive properties and generalizations that were largely unthought of in the absence of a way to say them. Sometimes these generalizations are quite simple: "Lexical NPs (proper nouns) denote monotone increasing functions". Others are deeper: "Lexical NP denotations are complete boolean generators for the full set of possible NP denotations". Chapters 5, 8, 10, and 11 here focus on such generalizations. Formulating them is exciting: looking at our world through mathematical lenses enables us to see vistas heretofore unsuspected. But of course, no sooner are new generalizations formulated than new empirical work challenges them, forcing refinements and extensions, so the half-life of a new generalization is relatively short. But this is the result of a healthy feedback relation between theory and observation.

We invite you to criticize our proposed generalizations and welcome you to Linguistics!

Helpful Hints on How to Read this Book Chapter 1 just introduces the reader to iterative processes in natural language. Learning a few good examples plus the form of argument that the set of expressions of English is infinite is all that is needed from this chapter. The material in Chapter 2 is used throughout the book, but it is standard, and readers with some mathematics background should read it quickly. Chapters 3 and 4 illustrate some basic syntactic formalization, broadly presupposed in Chapter 5. Chapter 5 itself, especially the second half, is a bit advanced for an introductory work and might be omitted on first reading. The chapter on formal language theory is largely self contained (though it uses the earlier formal work from Chapter 2). Chapters 7 through 11 are all semantic with each presupposing the previous ones, though Chapter 9 is relatively self-contained. Chapters 10 and 11 have fewer exercises than the others and focus more on empirical generalizations we can make using the mathematical notions we have developed.

Words of Thanks We thank our classes over the years for helping us to develop the material. Both authors would like to thank Uwe Moennich, Richard Oehrle, and Jeff Heinz for comments on a much earlier version of this manuscript. As well we both owe a debt to Nicolas LaCasse not only for the fine LaTeX presentation, but for several reorganizational changes that improved the clarity of presentation. In addition, Edward Keenan would like especially to thank his wife Carol Archie for having made it all possible and suffering through endless weekends of an apparently endless effort. She still thinks we should call the book Sex, Lies and Language. Larry Moss thanks Hans-Joerg Tiede for teaching a version of the material many years ago and for many discussions over the years about pedagogic issues in the area. He also thanks Madi Hirschland for her unfailing love and support, and for continuing to remind him that as beautiful as language and mathematics are, there are many other aspects of the world that need to be understood and even changed.

1

The Roots of Infinity

We begin our study with a fundamental query of modern Linguistics:

(1) How do speakers of a natural language produce and understand novel expressions?

Natural languages are ones like English, Japanese, Swahili, . . . which human beings grow up speaking naturally as their normal means of communication. There are about 5,500 such languages currently spoken in the world[1]. Natural languages contrast with *artificial languages* consciously created for special purposes. These include *programming languages* such as Lisp, C++ and Prolog and *mathematical languages* such as Sentential Logic and Elementary Arithmetic studied in mathematical logic. The study of natural language syntax and semantics has benefited much from the study of these much simpler and better understood artificial languages.

The crucial phrase in (1) is *novel*. An ordinary speaker is *competent* to produce and understand arbitrarily many expressions he or she has never specifically heard before and so certainly has never explicitly learned. This chapter is devoted to supporting this claim, introducing some descriptive mathematical notions and notations as needed.

In the next chapter we initiate the study of the linguists' response to the fundamental query: namely, that speakers have internalized a *grammar* for their language. That grammar consists of a set of *lexical items* – meaningful words and morphemes – and some *rules* which allow us to combine lexical items to form arbitrarily many complex expressions whose semantic interpretation is determined by that of the expressions they are formed from. We produce, recognize and interpret novel expressions by using our internalized grammar to recognize how

[1]This figure is "rough" for both empirical and conceptual reasons. For example, how different may two speech varieties be and still count as dialects of the same language as opposed to different languages?

the expressions are constructed, and how expressions constructed in that way take their meaning as a function of the meanings of what they are constructed from – ultimately the lexical items they are built from. This last feature is known as *Compositionality*.

In designing grammars of this sort for natural languages we are pulled by several partially antagonistic forces: Empirical Adequacy (Completeness, Soundness, Interpretability) on the one hand, and Universality on the other. Regarding the former, for each natural language L the grammar we design for L must be *complete*: it generates *all* the expressions native speakers judge grammatical; it must be *sound*: it *only* generates expressions judged grammatical, and it must be *interpretable*: the lexical items and derived expressions must be semantically interpreted. Even in this chapter we see cases where different ways of constructing the same expression may lead to different ways of semantically interpreting it. Finally, linguists feel strongly that the structure of our languages reflects the structure of our minds, and in consequence, at some deep level, grammars of different languages should share many structural properties. Thus in designing a grammar for one language we are influenced by work that linguists do with other languages and we try to design our (partial) grammars so that they are similar (they cannot of course be identical, since English, Japanese, Swahili, . . . are not identical).

1.1 The Roots of Infinity in Natural Language

Here we exhibit a variety of types of expression in English which support the conclusion that competent speakers of English can produce, recognize and understand unboundedly many expressions. What is meant by *unboundedly many* or *arbitrarily many*? In the present context we mean simply *infinitely many* in the mathematical sense. Consider for example the set $\mathbb{N}$ of natural numbers, that set whose members (elements) are the familiar 0, 1, 2, ... Clearly $\mathbb{N}$ has infinitely many members, as they "continue forever, without end". A less poetic way to say that is: a set L is *infinite* if and only if for each natural number k, L has more than k members. By this informal but usable definition we can reason that $\mathbb{N}$ is infinite: no matter what number k we pick, the numbers $0, 1, \ldots, k$ constitute more than k elements of $\mathbb{N}$; in fact precisely $k + 1$ elements. So for any k, $\mathbb{N}$ has more than k elements. This *proves* that $\mathbb{N}$ is an infinite set according to our definition of *infinite*.

Jargon In mathematical discourse *if and only if*, usually abbreviated *iff*, combines two sentences to form a third. P *iff* Q means that P and

Q always have the same truth value: in an arbitrary situation s they are both true in s or both false in s. *iff* is often used in definitions, as there the term we are defining occurs in the sentence on the left of *iff* and the sentence we use to define that term occurs on the right, and the purpose of the definition is to say that whenever we use the word being defined, we may replace it by the definition which follows.

When are sets the same? In what follows we shall often be interested in defining sets—for example, sets of English expressions with various properties. So it will be important to know when two apparently different definitions define the *same* set. We say that sets X and Y are *the same* iff they have the same members[2]. For example, let X be the set whose members are the numbers $1, 2, 3$, and 4. And let Y be the set whose members are the positive integers less than 5. Clearly X and Y have the same members and so are the same set. To say that X and Y are the same set we write $X = Y$ read as "X equals Y". And to say that X and Y are *different* sets we write $X \neq Y$, read as "X does not equal Y". Observe that $X \neq Y$ iff one of the sets X, Y has a member the other doesn't have. Were this condition to fail then X and Y would have the same members and thus be the same set ($X = Y$).

On sets and their sizes. The number of elements in a set X is noted $|X|$ and is called the *cardinality* of X. We first consider finite sets. A set X is *finite* iff for some natural number k, X has exactly k elements. That is, $|X| = k$. This definition is in practice useful and easy to apply. For example, the set whose elements are just the letters a, b, and c is finite, as it has exactly three elements. This set is usually noted $\{a, b, c\}$, where we list the names of the elements separated by commas, with the whole list enclosed in curly brackets (not angled brackets, not round brackets or parentheses). To say that an object x is a member (element) of a set A we write $x \in A$, using a stylized Greek epsilon to denote the membership relation. For example, $3 \in \{1, 3, 5, 7\}$. To say that x is not a member of A we write $x \notin A$. For example, $2 \notin \{1, 3, 5, 7\}$.

One finite set of special interest is the *empty* set, also called the *null* set, and noted $\emptyset$. It is that set with no members. Note that there could not be two different empty sets, for then one would have to have a member that the other didn't, so it would have a member and thus not be empty.

We have already mentioned that the set $\mathbb{N}$ of natural numbers is infinite. Sometimes we refer to it with a *pseudolist* such as $\{0, 1, 2, \ldots\}$

[2]In fact this criterion for identity of sets is one of the axioms (the *Axiom of Extensionality*) of Set Theory. It reads: For all sets A, B $A = B$ iff for all x, $x \in A$ iff $x \in B$.

where the three dots means simply that "the reader knows how to continue the list". This is a useful notation when in fact we know how to continue the list. But it does not count as a definition since there are many ways the initial explicitly given segment of the list, namely $0, 1, 2$, could be continued. Here we take it that in practice the reader has a working familiarity of the natural numbers. For example $2 \in \mathbb{N}$, but $\frac{1}{2} \notin \mathbb{N}$.

We demonstrate shortly that English has infinitely many expressions. To show this we consider three ways of showing that a set A is infinite. The first we have already seen: show that for every natural number k, A has more than k elements. This is how we convinced ourselves (if we needed convincing) that the set $\mathbb{N}$ of natural numbers was infinite. A second way uses the very useful subset relation, defined as follows:

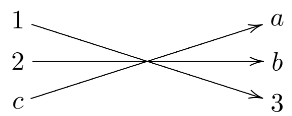
Definition 1.1. A set A is a *subset* of a set B, noted $A \subseteq B$, iff every element of A is also an element of B. More formally, $A \subseteq B$ iff for all objects x, if $x \in A$ then $x \in B$.

When $A \subseteq B$, it is possible that $A = B$, for then every element of A is, trivially, an element of B. But it is also possible that B has some element(s) not in A, in which case we say that A is a *proper subset* of B, noted $A \subset B$. Note too, that if A is not a subset of B, noted $A \nsubseteq B$, then there is some $x \in A$ which is not in B.

And we can now give our second "infinity test" as follows: a set B is infinite if it has a subset A which is infinite. The reasoning behind this plausible claim is as follows. If $A \subseteq B$ and A is infinite then for each natural number k, A has more than k elements. But each of those elements are also in B, so for any k, B has more than k elements and is thus infinite.

A last, and the most useful infinity test, is to show that a set B is infinite by showing that it has the *same cardinality* as a set (such as $\mathbb{N}$) which is already known to be infinite. The idea of sameness of cardinality is fundamental, and does not depend on any pretheoretical notion of natural number. We say that sets A and B have the same number of elements (the *same cardinality*) iff if we can match the elements of one with the elements of the other in such a way that distinct elements of one are always matched with distinct elements of the other, and no elements in either set are left unmatched. The matching in (2) shows that $A = \{1, 2, 3\}$ and $B = \{a, b, c\}$ have the same number of elements. The matching illustrated is one to one with nothing left over:

(2) $\underline{A}$ $\underline{B}$



Other matchings, such as 1 with b, 2 with c and 3 with a, would have worked just as well. A more interesting illustration is the matching given in (3), which shows that the set *EVEN*, whose elements are just 0, 2, 4, . . ., is infinite. To give the matching explicitly we must say what even number an arbitrary natural number n is matched with.

(3)
$$\begin{array}{cccccccc} \mathbb{N} & 0 & 1 & 2 & \cdots & n & \cdots \\ & \updownarrow & \updownarrow & \updownarrow & & \updownarrow & \\ \mathit{EVEN} & 0 & 2 & 4 & \cdots & 2n & \cdots \end{array}$$

So each natural number n is matched with the even number $2n$. And distinct n's get matched with distinct even numbers, since if n is different from m, noted $n \neq m$, then $2n \neq 2m$. And clearly no element in either set is left unmatched. Thus *EVEN* has the same size as $\mathbb{N}$ and so is infinite.

Exercise 1.1. This exercise is about infinite sets.

a. Show by direct argument that *EVEN* is infinite. (That is, show that for arbitrary k, *EVEN* has more than k elements).
b. Let *ODD* be the set whose elements are 1, 3, 5, ...
   a. Show directly that *ODD* is infinite.
   b. Show by matching that *ODD* is infinite.

We turn now to an inventory of types of expression in English whose formation shows that English has infinite subsets of expressions.

Iterated words There are a few cases in which we can build an infinite set of expressions by starting with some fixed expression and forming later ones by repeating a word in the immediately previous one. *GP* below is one such set; its expressions are matched with $\mathbb{N}$ showing that it is an infinite set.

(4)

$\mathbb{N}$	*GP*
0	my grandparents
1	my great grandparents
2	my great great grandparents
. . .	. . .
n	my (great)^n grandparents
. . .	. . .

In line n, (great)^n on the right denotes the result of writing down the word *great* n times in a row. Thus $great^1 = great$, $great^2 = great\ great$, etc.

When $n = 0$ we haven't written anything at all: so *my (great)*^{0} *grandparents* is simply *my grandparents*. We often leave spaces between words when writing sequences of words from a known language (such as English). We usually do not do this when concatenating random sequences of letters: $(aab)^3 = aabaabaab$.

One sees immediately from the matching in (4) that the set GP is infinite. Hence the set of English expressions itself is infinite since it has an infinite subset. Moreover the expressions in GP all have the same category, traditionally called *Noun Phrase* and abbreviated NP. So we can say that PH(NP), the set of phrases of category NP in English is infinite.

Note too that each expression in GP is meaningful in a reasonable way. Roughly, *my grandparents* denotes a set of four people, the (biological) parents of my parents. *my great grandparents* denotes the parents of my grandparents, an 8 element set; *my great great grandparents* denotes the parents of my great grandparents, a 16 element set, and in general the denotation of *my (great)*$^{n+1}$ *grandparents* denotes the parents of my (great)^n grandparents. (For each $n \in \mathbb{N}$, how many (great)^n grandparents do I have[3]?).

We turn in a moment to more structured cases of iteration which lead to infinite subsets of English. But first let us countenance one reasonable objection to our claim that *my (great)*^{n} *grandparents* is always a grammatical expression of English. Surely normal speakers of English would find it hard to to interpret such expressions for large n, even say $n = 100$, so can we not find some largest n, say $n = 1,000,000$, beyond which *my (great)*^{n} *grandparents* is ungrammatical English?

Our first response is a practical one. We want to state which sequences of English words are grammatical expressions of English. To this end we study sequences that are grammatical and ones that aren't and try to find regularities which enable us to predict when a novel sequence is grammatical. If there were only 25 grammatical expressions in English, or even several hundred, we could just list them all and be done with it. The grammaticality of a test expression would be decided by checking whether it is in the list or not. But if there are billions in the list that is too many for a speaker to have learned by heart. So we still seek to know on what basis the speaker decides whether to say Yes or No to a test expression. In practice, then, characterizing membership

[3]This is a tricky question.

in large finite sets draws on the same techniques used for infinite sets. In both cases we are looking for small descriptions of big sets.

Our task as linguists is similar to designing chess playing algorithms. Given some rules limiting repetitions of moves, the number of possible chess games is finite. Nonetheless we treat chess as a game in which possible sequences of moves are determined by rule, not just membership in some massive list.

The second response is that even for large n, speakers are reluctant to give a cut off point: $n = 7?$, $17?$, 10^{17} ? In fact we seem *competent* to judge that large numbers of repetitions of *great* are grammatical, and we can compute the denotation of the derived expression, though we might have to write it down and study it to do so. It is then like multiplying two ten digit numbers: too hard to do in our head but the calculation still follows the ordinary rules of multiplication. It seems reasonable then to say that English has some expressions which are too long or too complicated to be usable in practice (in *performance* as linguists say), but they are nonetheless built and interpreted according to the rules that work for simple expressions.

We might add to these responses the observation that treating certain sets as infinite is often a helpful simplifying assumption. It enables us to concentrate on the simple cases, already hard enough! We return now to the roots of infinity.

Postnominal Possessives Syntactically the mechanism by which we build the infinite set GP above is as trivial as one can imagine: arbitrary repetition of a single word, *great*. But English presents structurally less trivial ways of achieving similar semantic effects with the use of relation denoting nouns, such as *mother*, *sister*, *friend*, etc. Here is one such case, with the matching defined more succinctly than before:

(5) For each natural number n, let $M(n)$ be the result of writing down the sequence *the mother of* n times followed by *the President*. That is, $M(n) = (\textit{the mother of})^n\,\textit{the President}$.

Clearly M matches distinct numbers n and n' with different English expressions since $M(n)$ and $M(n')$ differ with regard to how many times the word *mother* occurs: in $M(n)$ it occurs n times, and $M(n')$ it occurs n' times. Clearly then the set whose elements are $M(0)$, $M(1)$, ... is an infinite set of English NPs .

Moreover, what holds syntactically holds semantically: when $n \neq n'$, $M(n)$ *and* $M(n')$ denote different objects. Think of the people $M(n)$ arranged in a sequence $M(0)$, $M(1)$, $M(2)$, ... = *the President, the mother of the President, the mother of the mother of the President, ...*. Now think of the sequence of denotations y_0, y_1, ..., they determine.

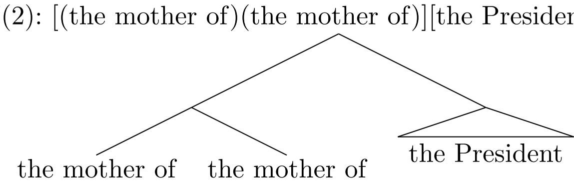
the President denotes some individual y_0, and each later expression, *(the mother of)^k the President*, denotes an individual y_k who is the mother of the immediately preceding individual y_{k-1} . Since no one can be their own (biological) mother, grandmother, greatgrandmother, etc., all these individuals y_i are different. So y_n, the denotation of *(the mother of)^n the President*, is different from $y_{n'}$, the denotation of *(the mother of)$^{n'}$ the President*.

Exercise 1.2.

a. Exhibit the matching in (5) using the format in (4).
b. Like *great* above, *very* can be repeated arbitrarily many times in expressions like *He is tall, He is very tall, He is very very tall,. . .*. Define a matching using any of the formats so far presented which shows that the number of expressions built from *very* in this way is infinite.

Observe that the matching function M introduced in (5) has in effect imposed some *structure* on the expressions it enumerates. For any $n > 0$, $M(n)$ is an expression which consists of two parts, (*constituents*), namely *the mother of*, written n times, and *the President*. And the leftmost constituent itself consists of n identical constituents, *the mother of*. We exhibit this structure for $M(1)$ and $M(2)$ below. (6c) is a *tree* representation of the constituent structure of $M(2)$.

(6) a. $M(1)$: [the mother of][the President]
b. $M(2)$: [(the mother of)(the mother of)][the President]
c.

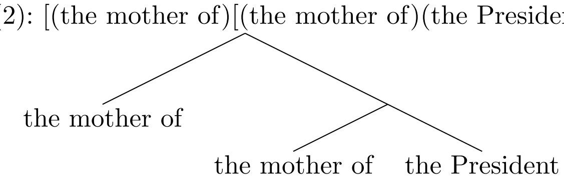


Now $M(1)$ is not a constituent of $M(2)$. Therefore given Compositionality, the meaning of $M(1)$ is not part of the meaning of $M(2)$. This seems slightly surprising, as one feels that $M(2)$ denotes the mother of the individual denoted by $M(1)$. For example, if Martha is the mother of the President then $M(2)$ denotes the same as *the mother of Martha*. So let us exhibit a different, *recursive*, way of enumerating the expressions in (5) in which $M(1)$ is a constituent of $M(2)$.

(7) For each $n \in \mathbb{N}$, $F(n)$ is a sequence of English words defined by:
a. $F(0) =$ *the President*,
b. For every $n \in \mathbb{N}$, $F(n + 1) =$ *the mother of* $F(n)$.

(Note that F succeeds in associating an expression with each number since any number k is either 0 or the result of adding 1 to the previous number). Observe too that M and F associate the sequence of English words with same number (try some examples!). So, for each n, $F(n) = M(n)$. But F and M effect this association in different ways, ones which will be reflected in the semantic interpretation of the expressions. Compare $F(2)$ with $M(2)$. $F(2)$ has two constituents: *the mother of* and $F(1)$, the latter also having two constituents, *the mother of* and *the President*. (8) exhibits the constituent structure imposed by F in $F(1)$ and $F(2)$. The gross constituent structure for $F(2)$ is given by the tree in (8c).

(8) a. $F(1)$: [the mother of][the President]
b. $F(2)$: [(the mother of)[(the mother of)(the President)]
c.



On the analysis imposed by F, $F(1)$ is a constituent of $F(2)$, and in general $F(n)$ is a constituent of $F(n+1)$. So $F(n)$ will be assigned a meaning in the interpretation of $F(n+1)$.

Exercise 1.3. Exhibit the constituent structure tree for $F(3)$ analogous to that given for $F(2)$ in (8c).

So these two analyses, M and F, make different predictions about what the meaningful parts of the expressions are[4]. Our original suggestion for an F-type analysis was that the string $M(1)$ – a *string* is just a sequence of symbols (letters, words, ...) – was a meaningful part of $M(2)$. Notice that the fact that the string $M(1)$ occurs as a *substring* of $M(2)$ was not invoked as motivation for an F-type analysis. And this is right, as it happens often enough that a substring of an expression is, accidentally, an expression in its own right but not a constituent of the original. Consider (9):

(9) The woman who admires John is easy to please.

Now (10a) is a substring of (9), and happens to be a sentence in its own right with the same logical meaning as (10b) and (10c).

(10) a. John is easy to please.
b. To please John is easy.

[4]In fact both analyses are amenable to a compositional semantic analysis, but the analyses differ and the M one requires a richer semantic apparatus than the F one.

c. It is easy to please John.

But any attempt to replace *John is easy to please* in (9) by the strings in (10b) or (10c) results in ungrammaticality (as indicated by the asterisk):

(11) a. *The woman who admires to please John is easy.
b. *The woman who admires it is easy to please John.

The reason is that *John is easy to please* is not a constituent, a meaningful part, of (9).

The constituent structure trees for the F analysis in (8) are *right branching* in the sense that as n increases, the tree for $F(n)$ has more nodes going down the righthand side. Compare the trees in Figure 1.

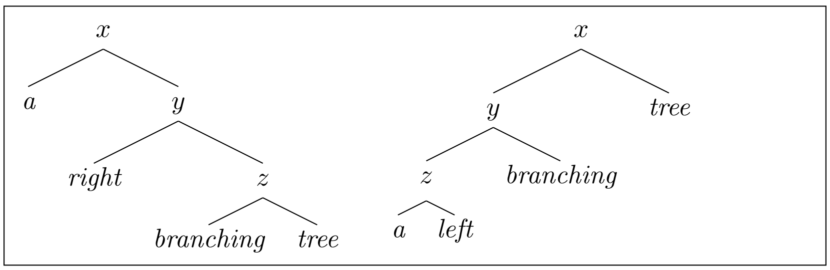


FIGURE 1 Two trees.

Verb final languages, such as Turkish, Japanese, and Kannada (Dravidian; India) are usually Subject+Object+Verb (SOV) and favor right branching structures: [John [poetry writes]]. By a very slight margin SOV is the most widespread order of Subject, Object and Verb across areas and genetic families (Baker (2001) pg. 128) puts this type at about 45% of the world's languages. Its mirror image, VOS, is clearly a minority word order type, accounting perhaps for 3% of world languages. It includes Tzotzil (Mayan, Mexico), Malagasy (Austronesian; Madagascar) and Fijian (Austronesian; Fiji). VOS languages prefer left branching structures: [[writes poetry] John]. The second most common word order type, SVO, like English, Swahili, Indonesian, accounts for about 42% of the world's languages, and VSO languages, such as Classical Arabic, Welsh, and Maori (Polynesian; New Zealand) account for perhaps 9%. The branching patterns in SVO and VSO languages are somewhat mixed, though on the whole they behave more like VOS languages than like SOV ones. See Dryer (2007) pp. 61–131 for excellent discussion of the complexities of the word order classification.

Finally we note that our examples of postnominal possessive NPs in (8) are atypically restricted both syntactically and semantically. It is unnatural for example to replace the Determiner *the* with others like *every, more than one, some, no, ...*. Expressions like *every mother of the President, more than one mother of the President* seem unnatural. This is likely due to our understanding that each individual has a unique (exactly one) biological mother. Replacing *mother* by less committal relation nouns such as *friend* eliminates this unnaturalness. Thus the expressions in (12) seem natural enough:

(12) *the President*, *every friend of the President*,
every friend of every friend of the President, ...

Now we can match the natural numbers with the elements of the pseudolist in (12) in a way fully analogous to that in (5). Each n gets matched directly with $($*every friend of*$)^n$ followed by *the President*.

Exercise 1.4. Exhibit a matching for (12) on which *every friend of the President* is a constituent of *every friend of every friend of the President*.

This has been our first instance of formalizing the same phenomena in two different ways (M and F). In fact being able to change your formal conceptualization of a phenomenon under study is a major advantage of mastering elementary mathematical techniques. Formulating an issue in a new way often leads to new questions, new insights, new proofs, new knowledge. As a scientist what you can perceive and formulate and thus know, is limited by your physical instrumentation (microscopes, lab techniques, etc.) and your mental instrumentation (your mathematical concepts and methods). *Man sieht was man weiss* (*One sees what one knows*). Mathematical adaptability is also helpful in distinguishing what is fundamental from what is just notational convention. The idea here is that significant generalizations are ones that remain invariant under changes of descriptively comparable notation. Here the slogan is:

If you can't say something two ways you can't say it at all.

Worth emphasizing here also is that mathematically we often find different procedures (algorithms) that compute the same value for the same input, but do so in different ways. Here is an example from high school algebra: compare the functions f and g below which map natural numbers to natural numbers:

(13) a. $f(n) = n^2 + 2n + 1$

b. $g(n) = (n+1)^2$

g here seems to be a simple two step process: given n, add 1 to it and square the result. f is more complicated: first square n, hold it in store and form another number by doubling n, then add those two numbers, and then add 1 to the result. But of course from high school algebra we know that for every n, $f(n) = g(n)$. That is, these two procedures always compute the same value for the same argument, but they do so in different ways. A cuter, more practical, example is given in the exercise below.

Exercise 1.5. Visitors to the States must often convert temperature measured on the Fahrenheit scale, on which water freezes at 32 degrees and boils at 212, to Celsius, on which water freezes at 0 degrees and boils at 100. A standard conversion algorithm C takes a Fahrenheit number n, subtracts 32 and multiplies the result by 5/9. So for example $C(212) = (212 - 32) \times 5/9 = 180 \times 5/9 = 20 \times 5 = 100$ degrees Celsius, as desired. Here is a different algorithm, C'. It takes n in Fahrenheit, adds (!) 40, multiplies by 5/9, and then subtracts 40. Show that for every natural number n, $C(n) = C'(n)$.

Prenominal possessors are possessive NPs like those in (14a), enumerated by G in (14b).

(14) a. the President's mother, the President's mother's mother, the President's mother's mother's mother, ...
b. $G(0) =$ *the President*, and for all $n \in \mathbb{N}$, $G(n+1) = G(n)$*'s mother*.

So $G(2) = G(1)$'s mother $= G(0)$'s mother's mother $=$ the President's mother's mother. This latter NP seems harder to process and understand than its right branching paraphrase *the mother of the mother of the President*. We have no explanation for this.

Adjective stacking is another left branching structure in English, easier to understand than iterated prenominal possessives but ultimately more limited in productivity. The easy to understand expressions in (15) suggest at first that we can *stack* as many adjectives in front of the noun as we like.

(15) a. a big black shiny car
b. an illegible early Russian medical text

But attempts to permute the adjectives often lead to less than felicitous expressions, sometimes gibberish, as in **a medical Russian early illegible text*. Now if we can't permute the adjectives, that suggests that adjectives come in classes with fixed positions in relation to the noun they modify, whence once we have filled that slot we can no longer add adjectives from that class, so the ability to add more is reduced. And this appears to be correct (Vendler). We can substitute other nationality adjectives for *Russian* in (15b), as in *an illegible early Egyptian medical text*, but we cannot add further nationality adjectives in front of *illegible*, **an Egyptian illegible early Russian medical text*. It is plausible then that there is some n such that once we have stacked n adjectives in front of a noun no further ones can be added. If so, the number of adjective-noun combinations is finite. In contrast, postnominal modification by relative clauses seems not subject to such constraints:

Relative clauses have been well studied in modern linguistics. They are illustrated by those portions of the following expressions beginning with *that*:

(16)
a. This is the house that Jack built.
b. This is the malt that lay in the house that Jack built.
c. This is the rat that ate the malt that lay in the house that Jack built.
d. This is the cat that killed the rat that ate the malt that lay in the house that Jack built.

These examples of course are adapted from a child's poem, and suggest that relative clauses can iterate in English: for each natural number n we can construct a relative clause with more than n properly nested relative clauses, whence the number of such clauses is infinite, as is the number of NPs which contain them. One might (rightly) quibble about this quick argument from the above example, however, on the grounds that successively longer relative clauses obtained as we move down the list use words *not in the previous sentences*. So if the number of words in English were finite then perhaps the possibility of forming novel relative clauses would peter out at some point, albeit a very distant point, as even desk top dictionaries in English list between $100,000$ to $150,000$ words. But in fact this is not a worry, as we *can* repeat words and thus form infinitely many relative clauses from a small (finite) vocabulary:

(17) Let

$$\begin{array}{lcl} H(0) & = & \textit{every student}\text{, and for all } n \in \mathbb{N},\text{ and} \\ H(n+1) & = & \textit{every student who knows } \mathrm{H(n)}. \end{array}$$

Thus

$H(1)$	$=$	*every student who knows H(0)*
	$=$	*every student who knows every student*
$H(2)$	$=$	*every student who knows H(1)*
	$=$	*every student who knows every student who knows every st.*

And so on. Clearly H enumerates an infinite set of NPs built from a four-word vocabulary: *every, student, who, knows*.

The relative clauses in these examples consist of a *relative pronoun*, *that* or *who*, followed by a sentence with an NP missing. In the cases that iterate in (16) it is always the *subject* that is missing. Thus in *the malt that lay in the house* ... *that*, which refers back to *malt*, is the subject of *lay in the house* ...; in *the rat that ate the malt* ..., *that* is the subject of *ate the malt*..., and in *the cat that killed the rat*..., *that* is the subject of *killed the rat* .... In the rightmost relative clause, *that Jack built*, *that*, which refers to *house*, is the *object* of the transitive verb *build*. Notice that the matching function H in (17) provides a right branching structure for the NPs enumerated. This analysis, naively, follows the semantic interpretation of the expressions. Viz., in $H(n+1)$, $H(n)$ is a constituent, in fact an NP, the sort of expression we expect to have a meaning. But as we increase n, the intonationally marked groups are different, an intonation peak being put on each noun that is later modified by the relative clause, as indicated by square bracketing in (1.1) in which each right bracket] signals a slight pause and the noun immediately to its left carries the intonation peak:

This is the cat [that killed the rat][that ate the malt] ...

A matching function H' that would reflect this bracketing in (1.1) would be: $H'(n) =$ *every student (who knows every student)*^{n}.

Attempts to iterate object relatives rather than subject ones quickly lead to comprehension problems, even when they are right peripheral. (18a) is an English NP built from an object relative. It contains a proper noun *Sonia*. (18b) is formed by replacing *Sonia* with another NP built from an object relative. Most speakers do not really process (18b) on first pass, and a further replacement of *Sonia* by an object relative NP yielding(18c) is incomprehensible to everyone (though of course you can figure out what it means with pencil and paper).

(18)
   a. some student who Sonia interviewed.
   b. some student who some teacher who Sonia knows interviewed.
   c. some student who some teacher who every dean who Sonia dislikes knows interviewed

The comprehensibility difference between the iterated object relatives in (18) and iterated subject ones in (16) is quite striking. Linguists have suggested that the reason is that the iteration of object relatives leads to *center embedding*: we replace a Y in a string with something that contains another Y but also material to the left and to the right of that new Y, yielding XYZ. So the new Y is center-embedded between X and Z. One more iteration yields a string of the form $[X_1[X_2YZ_2]Z_1]$, and in general $n+1$ iterations yields

$$[X_1 \ \ldots [X_n \ [X_{n+1} \ Y \ Z_{n+1} \ ]Z_n] \ \ldots \ Z_1].$$

Postnominal Prepositional Phrase (PP) modification is illustrated by NPs of the form [$_{\text{NP}}$ Det N [$_{\text{PP}}$P NP]], such as *a house near the port*, *two pictures on the wall*, *a doctor in the building*. Choosing the rightmost NP to be one of this form, [Det N [P NP]], we see the by now familiar right branching structure. Again a children's song provides an example of such iterated PPs :

(19) a. There's a hole in the bottom of the sea.
b. There's a log in the hole in the bottom of the sea.
c. There's a bump on the log in the hole in the bottom of the sea.
d. There's a frog on the bump on the log in the hole in the bottom of the sea.

As with the relative clauses we must ask whether we can iterate such PPs without always invoking new vocabulary. We can, but our examples are cumbersome:

(20) a. a park near [the building by the exit]
b. a park near [the building near [the building by the exit]]
c. a park (near the building)^n by the exit

Note that one might argue that PP modification of nouns is not independent of relative clause modification on the grounds that the grammar rules of English will derive the PPs by *reducing* relative clauses: *a house near the port* $\Leftarrow$ *a house which is near the port*. Perhaps. If so, then we have just shown that such reduction is not blocked in contexts of multiple iteration.

Sentence complements concern the objects of verbs of thinking and saying such as *think, say, believe, know, acknowledge, explain, imagine, hope, etc.*. They would be most linguists' first choice of an expression type which leads to infinitely many grammatical expressions, as shown in (21):

(21) $SC(0)$ = he destroyed the house;

$SC(n+1) =$ he said that $SC(n)$.

SC enumerates in a right branching way *he said that he said that ... he destroyed the house*. Note too that such expressions *feed* relative clause formation: *the house which (he said that)^n he destroyed*, yielding another infinite class of NPs .

Sentential subjects in their simplest form resemble sentence complements but function as the subject of a verb rather than as the objects. They determine center embeddings, and as with the earlier cases become virtually impossible to understand after one embedding:

(22) a. That Sue quit surprised me.
b. That that Sue quit surprised me is strange.
c. That that that Sue quit surprised me is strange is false.

Even (22b) is sufficiently hard to understand that we perhaps should just consider it ungrammatical. However if the sentential subjects are replaced by sentence complements of nouns, such as *the claim that Sue left early*, the resulting Ss improve:

(23) a. The claim that Sue quit surprised me.
b. The belief that the claim that Sue quit surprised me is strange.
c. The fact that the belief that the claim that Sue quit surprised me is strange is really outrageous.

Here (23b) can be given an intonation contour that makes it more or less intelligible. Another variant of (22) concerns cases in which the sentential subject has been *extraposed*, as in (24).

(24) a. It surprised me that Sue quit.
b. It is strange that it surprised me that Sue quit.
c. It is false that it is strange that it surprised me that Sue quit.

These are right branching expressions and are considerably easier to comprehend than their center embedding paraphrases in (22).

Caveat Lector On the basis of (22) and (18) we are tempted to conclude that center embedding in general is difficult to process. Don't! One robin doesn't make a spring and English is but one of the 5,500 languages in the world. SOV languages present a variety of expression types which induce center embedding. These types include some that translate the right branching sentence complements in English. Consider for example sentence (25) from Nepali:

(25) *Gītāle* *Rāmlāī* *Anjalīlāī* *pakāuna* *sahayog garna*
Gita Ram Anjali cook help
sallāh garī.
advised.

Gita advised Ram to help Anjali cook.

In these, *lāī* is a postposition carried by human names which are objects of transitive verbs, and *le* is a postposition associated with the past tense form. The forms *pakāuna* and *sahayog garna* are infinitives. As one can see from this example, the subjects are grouped together ahead of all of the verbs. The form is the center embedding pattern

$$\mathrm{NP}_1\ \mathrm{NP}_2\ \mathrm{NP}_3\ \mathrm{V}_3\ \mathrm{V}_2\ V_1,$$

with two proper center embeddings, rather than the (rough) form of its English translation:

$$\mathrm{NP}_1\ \mathrm{V}_1\ \mathrm{NP}_2\ \mathrm{V}_2\ \mathrm{NP}_3\ \mathrm{V}_3.$$

(Again, the subjects and verbs, whether in the main clause or in embedded clauses, have special endings that need not concern us.)

Now, impressionistically, Nepali speakers seem to process Ss like (25) easily, arguing against rash conclusions concerning the difficulty of center embedding. More psycholinguistic work is needed.

Formally though the existence of unbounded center embedding would remove languages that have it from the class called *regular* those recognizable by finite state machines (the lowest class in the Chomsky hierarchy). See Hopcroft and Ullman 1979:Ch 9 for the relevant definitions and theorems.

Exercise 1.6.

a. Exhibit an enumeration of infinitely many Ss of the sort in (18) allowing repetition of verbs, as in *that he quit surprised me, that that he quit surprised me surprised me, ...*
b. Exhibit an enumeration of their extraposed variants, (24), under the same repetition assumptions.

Infinitival complements come in a variety of flavors in English illustrated in (26–28). The infinitival complements in (26) and (27) are the untensed verbs preceded by *to*. Verbs like *help, make, let* and perception verbs like *watch, see, hear* take infinitival complements without the *to*.

(26) a. Mary wanted to read the book.
b. She wanted to try to begin to read the book.

(27) a. She asked Bill to wash the car.

b. She forced Joe to persuade Sue to ask Sam to wash the car.

(28) a. She helped Bill wash the car.
b. She let Bill watch Harry make Sam wash the car.

The b-sentences suggest that we can iterate infinitival phrases, though the intransitive types in (26) are hard to interpret. What does *He began to begin to begin to wash the car* mean? The transitive types in (27) and (28), where the verb which governs the infinitival complement takes an NP object, iterate more naturally: *She asked Bill to ask Sue to ask Sam to ... to wash the car. She helped Bill help Harry help Sam ... wash the car* or *She watched Bill watch Mary watch the children ... play in the yard*. Repeating proper names here, however, can be unacceptable: *?*She asked Bill to ask Bill to wash the car*. But to show that this structure type leads to infinitely many expressions it suffices to choose a *quantified* NP object, as in (29).

(29) a. She asked a student to ask a student to ... to wash the car.
b. She helped a student help a student ... wash the car.

Exercise 1.7.

a. Exhibit an enumeration of one of the sets in (29).
b. Another way to avoid repeating NP objects of verbs like *ask* is to use a previously given enumeration of an infinite class of NPs. Exhibit an enumeration of *She asked every friend of the President to wash the car, She asked every friend of the President to ask every friend of every friend of the President to wash the car, ...*.

Embedded questions are similar to sentence complements, but the complement of the main verb semantically refers to a question or its answer, not a statement of the sort that can be true or false. Compare (30a), a True/False type assertion, with (30b), which requests an identification of the *Agent* of the verb *steal*, and (30c), an instance of an embedded question, where in effect we are saying that John knows the answer to the question in (30b).

(30) a. Some student stole the painting.
b. Which student stole the painting?
c. John knows which student stole the painting.

The question in (30b) differs from the assertion in (30a) by the choice of an interrogative Determiner *which* as opposed to *some*[5]. The embed-

[5]This simple relation between declaratives and interrogatives only holds when we question the subject. In other types of constituent questions the interrogative expression is moved to the front of the clause and the subject is moved behind the auxiliary verb if there is one; if there isn't an appropriately tensed form of *do* is

ded question following *knows* in (30c) uses the same interrogative Det. And of course we can always question the subject constituent of sentences like (30c), yielding Ss like (31a), which can in turn be further embedded, *ad infinitum*.

(31) a. Who knew which student stole the painting?
b. John figured out who knew which student stole the painting
c. Which detective figured out who knew which student stole the painting?
d. John knew which detective figured out who knew which student stole the painting

In distinction to sentence complements, attempts to form relative clauses from embedded questions lead to expressions of dubious grammaticality (indicated by ?? below):

(32) ?? the painting that John knew which detective figured out which student stole

Exercise 1.8. The function EQ exhibited below (note the notation) matches distinct numbers with distinct expressions, so the set of embedded questions it enumerates is infinite.

$EQ \qquad n \mapsto$ *Joe found out (who knew)*^{n} *who took the painting.*

Your task: Exhibit a recursive function EQ' which effects the same matching as EQ but does so in such a way as to determine a right branching constituent structure of the sort below:

Joe found out [who knew [who knew [who took the painting]]].

Notation and a Concept The matchings we have exhibited between $\mathbb{N} = \{0, 1, 2, \ldots\}$ and English expressions have all been *one-to-one*, meaning that they matched distinct numbers with distinct expressions. More generally, suppose that f is a function from a set A to a set B, noted $f : A \to B$. A is called the *domain* of the function f, and B is called its *codomain*. So f associates each object $x \in A$ with a unique object $f(x) \in B$. $f(x)$ is called *the value* of the function f at x. We also say that f *maps* x to $f(x)$. "Unique" here just means "exactly one"; that is, f maps each $x \in A$ to some element of B, and f does not map any x to more than one element of B. Now, f is said to be *one-to-one* (synonym: *injective*) just in case f maps distinct $x, x' \in A$ to distinct elements $f(x), f(x') \in B$[6]. So a one-to-one function is one

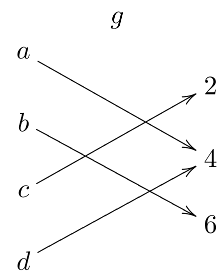
inserted. Compare: *John stole some painting* with *Which painting did John steal?*

[6] A one-to-one function is actually "two-to-two."

that preserves distinctness of arguments (the elements of A being the *arguments* of the function f). A function which fails to be one-to-one fails to preserve distinctness of arguments, so it must map at least two distinct arguments to the same value.

We also introduce some notation that we use in many later chapters. Let $[A \to B]$ denote the set of functions from A to B.

Exercise 1.9. In the diagram below we exhibit in an obvious way a function g from $A = \{a, b, c, d\}$ to $B = \{2, 4, 6\}$. The use of arrows tells us that the elements at the tail of the arrow constitute the domain of g, and those at the head lie in the codomain of g.



a. Is g one-to-one? Justify your answer.
b. Do there exist any one-to-one functions from A into B?
c. Exhibit by diagram two different one-to-one functions from B into A.

Returning now to functions, we say that a set A is less than or equal in size (cardinality) to a set B, written $A \preceq B$, iff there is a one-to-one map from A into B. (We use *map* synonymously with *function*). This (standard) definition is reasonable: if we can copy the elements of A into B, matching distinct elements of A with distinct elements of B, then B must be at least as large as A.

Observe that for any set A, $A \preceq A$. *Proof:* Let id be that function from A into A given by: $id(\alpha) = \alpha$, for all $\alpha \in A$[7]. So trivially id is injective (one-to-one).

Exercise 1.10.

a. Let A and B be sets with $A \subseteq B$. Show that $A \preceq B$.
b. Show that $EVEN \preceq \mathbb{N}$, where, $EVEN = \{0, 2, 4, \ldots\}$, and $\mathbb{N} = \{0, 1, 2, \ldots\}$.

[7] α is the first "lower case" letter of the Greek alphabet, given in full at the end of this text. The reader should memorize this alphabet together with the names of the letters in English, as Greek letters are widely used in mathematical discourse.

c. Show that $\mathbb{N} \preceq EVEN$.

Mathematically we define sets A and B to have the same size (cardinality), noted $A \approx B$, iff $A \preceq B$ and $B \preceq A$. A famous theorem of set theory (the Schroeder-Bernstein Theorem) guarantees that this definition coincides with the informal one—matching with nothing left unmatched—given earlier. On this definition, then, $EVEN \approx \mathbb{N}$, that is, the set $\{0, 2, 4, \ldots\}$ of natural numbers that are even has the same size as the whole set $\mathbb{N}$ of natural numbers. Using the cardinality notation introduced earlier we see that $A \approx B$ iff $|A| = |B|$. So for example, $|EVEN| = |\mathbb{N}|$. Can you show that $\mathbb{N} \approx ODD$, the set $\{1, 3, 5, \ldots\}$ of natural numbers that are odd?

Here are two useful concepts defined in terms of $\preceq$. First, we say that a set A is *strictly smaller* than a set B, noted $A \prec B$, iff A is less than or equal in size to B and B is not less than or equal in size to A. That is, $A \prec B$ iff $A \preceq B$ and $B \npreceq A$. For example, $\{a, b\} \prec \{4, 5, 6\}$. An alternative way to say $A \prec B$ is $|A| < |B|$.

One might have expected that $|EVEN| < |\mathbb{N}|$ given that $EVEN \subset \mathbb{N}$ (that is $EVEN$ is a proper subset of $\mathbb{N}$). But that is not the case. The map sending each even number k to itself is a one-to-one map from $EVEN$ into $\mathbb{N}$, so $EVEN \preceq \mathbb{N}$. And the map sending each natural number n to $2n$ is one-to-one from $\mathbb{N}$ into $EVEN$, so $\mathbb{N} \preceq EVEN$, whence $|EVEN| = |\mathbb{N}|$

Occasionally students find it "unintuitive" that $EVEN$ and $\mathbb{N}$ have the same cardinality, given that $EVEN$ is a very proper subset of $\mathbb{N}$. $\mathbb{N}$ after all contains infinitely many numbers not in $EVEN$, namely all the odd numbers (ones of the form $2n + 1$). Here is an informal way to appreciate the truth of $EVEN \approx \mathbb{N}$. Consider that a primary purpose of $\mathbb{N}$ is to enable us to count any finite set. We do this by associating a distinct natural number with each of the objects we are counting, Usually the association goes in order, starting with 'one'. Then we point to the next object saying 'two', then the next saying 'three', etc. until we are done. $\mathbb{N}$ in fact is the "smallest" set that enables us to count any finite set in this way. But we could make do with just the set of even numbers. Then we would count 'two', 'four', 'six', etc. associating with each object from our finite set a distinct even number. So any set we can count with $\mathbb{N}$ we can count Edith $EVEN$, since they are equinumerous.

This relation between $EVEN$ and $\mathbb{N}$ provides a conceptually pleasing (and standard) way to define the notions *finite* and *infinite* without reference to numbers. Namely, an infinite set is one which has the same cardinality as some proper subset of itself. So a set A is *infinite* iff there is a proper subset B of A such that $A \approx B$. And A is *finite* iff A is not infinite (so A does not have the same size as any proper subset of itself). These definitions identify the same sets as we did earlier when we said that a set A is infinite iff for every natural number k, A has more than k elements. And A is finite iff for some natural number k, A has exactly k elements. But these earlier (and quite useful) definitions do rely on prior knowledge of the natural numbers. Our later ones above characterize *finite* and *infinite* just in terms of matchings—namely, in terms of one-to-one functions

We return now to a final case of structure building operations in English.

1.2 Boolean Compounding

Boolean compounding differs from our other expression building operations which build expressions of a fixed category. The *boolean connectives* however are *polymorphic*, they combine with expressions of many different categories to form derived expressions in that same category. The boolean connectives in English are *(both) … and, (either) … or, neither … nor …, not* and some uses of *but*. Below we illustrate some examples of boolean compounds in different categories. We label the categories traditionally, insisting only that expressions within a group have the same category. We usually put the examples in the context of a larger expression, italicizing the compound we are illustrating.

(33) Boolean Compounds in English

a. Noun Phrases *Neither John nor any other student* came to the party. *Most of the students and most of the teachers* drink. *Not a creature* was stirring, *not even a mouse.*

b. Verb Phrases
She *neither sings nor dances*, He *works in New York and lives in New Jersey*, He *called us but didn't come over.*

c. Transitive Verb Phrases
John *both praised and criticized* each student, He *neither praised nor criticized* any student, He *either admires or believes to be a genius* every student he has ever taught.

d. Adjective Phrases
This is an *attractive but not very well built* house. He is *neither intelligent nor industrious.* She is a *very tall and very graceful* dancer

e. Complementizer Phrases
He believes *neither that the Earth is flat nor that it is*

round. He believes *that it is flat but not that it is heavy*. He showed *either that birds dream or that they don't*, I forget which,

f. Prepositional Phrases
That troll lives *over the hill but not under a bridge*. A strike must pass *above the batter's knees and below his chest*.

g. Prepositions
There is no passage *either around or through* that jungle. He lives *neither in nor near* New York City.

h. Determiners
Most but not all of the cats were inoculated. *At least two but not more than ten* students will pass. *Either hardly any or else almost all* of the students will pass that exam.

i. Sentences
John came early and Fred stayed late. *Either John will come early or Fred will stay late. Neither did any student attend the lecture nor did any student jeer the speaker.*

In terms of productivity, boolean compounds are perhaps comparable to iterating adjectives: we can do it often, but there appear to be restrictions on repeating words which would mean that the productivity of boolean compounding is bounded. There are a few cases in which repetition is allowed, with an intensifying meaning:

(34) John laughed, and laughed, and laughed.

But even here it is largely unacceptable to replace *and* by *or* or *neither ... nor ...*: **John either laughed or laughed or laughed.* Equally other cases of pure repetition seem best classed as ungrammatical:

(35) *Either every student or every student came to the party. *Fritz is neither clever nor clever. *He lives both in and in New York City.

On the other hand judicious selection of different words allows the formation of quite complex boolean compounds, especially since *and* and *or* combine with arbitrarily many (distinct) expressions, as per (36b):

(36) a. either John and his uncle or else Bill and his uncle but not Frank and his uncle or Sam and his uncle
b. John, Sam, Frank, Harry and Ben but not Sue, Martha, Rosa, Felicia or Zelda

Note too that the polymorphism of the boolean connectives allows the formation of Ss for example with boolean compounds in many categories simultaneously:

(37) Every student, every teacher and every dean drank and caroused in some nightclub or bar on or near the campus until late that night or very early the next morning

Concluding remarks. We have exemplified a variety of highly productive ways of forming English expressions. This led to the conclusion that English has infinitely many expressions. Note though–an occasional point of confusion—each English expression itself is finite (just built from finitely many words). Our claim is only that there are infinitely many of these finite objects.

We also raised the general linguistic challenge of accounting for how speakers of English (or any natural language) produce, recognize and interpret large numbers of novel expressions. Our general answer was that speakers have learned a grammar and learned how to compositionally interpret the infinitely many expressions it generates. In the remainder of this book we concern ourselves with precisely how to define such grammars and how to interpret the expressions they generate. In the process we shall enrich the mathematical apparatus we have begun to introduce here. And as with many mathematics-oriented books, much of the learning takes place in doing the exercises. If you only read the text without working the exercises you will miss much.

Learning mathematics is like learning to dance: You learn little just by watching others.

1.3 References

Center-embedding was the subject of numerous papers in the linguistic and psychological literature. Some of the earliest references are Chomsky and Miller (1963), Miller and Isard (1964), and de Roeck et al. (1982).

Since 1990 the subject of center-embedding has again been taken up by researchers interested in processing models of human speech. Some references here are Church (1980), Abney and Johnson (1991), and and Resnik (1992).

A paper presenting psycholinguistic evidence that increasing center embedding in SOV languages does increase processing complexity is Babyonyshev and Gibson (1999).

2

Some Mathematical Background

2.1 More about Sets

The terms *boolean connective* and *boolean compound* derive more from logic than linguistics and are based on the (linguistically interesting) fact that expressions which combine freely with these connectives are semantically interpreted as elements of a set with a *boolean structure*. We use this structure extensively throughout this book. Here we exhibit a "paradigm case" of a boolean structure, without, yet, saying fully what it is that makes it boolean. Our example will serve to introduce some further notions regarding sets that will also be used throughout this book.

Consider the three element set $\{a, b, c\}$. Call this set X for the moment. X has several subsets. For example the one-element set $\{b\}$ is a subset of X. We call a one-element set a *unit set*. Recall that this fact is noted $\{b\} \subseteq X$. This is so because every element of $\{b\}$ – there is only one, b – is an element of X. Similarly the other unit sets, $\{a\}$ and $\{c\}$, are both subsets of X. Equally there are three two-element subsets of X: $\{a, b\}$ is one, that is, $\{a, b\} \subseteq X$. What are the other two? And of course X itself is a subset of X, since, trivially, every object in X is in X. (Notice that we are *not* saying that X is an *element* of itself, just a subset.) There is one further subset of X, the *empty set*, $\emptyset$. This set was introduced on page 3. Recall that $\emptyset$ has no members. Trivially $\emptyset$ is a subset of X (indeed of any set). Otherwise there would be something in $\emptyset$ which isn't in X, and there isn't anything whatsoever in $\emptyset$.

The set of subsets of a set X is called *the power set* of X and noted $\mathcal{P}(X)$. In the case under discussion we have:

$$\mathcal{P}(\{a, b, c\}) \quad = \quad \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}.$$

On the right we have a single set with 8 elements; those elements are themselves all sets. Indeed, on the right we have just listed all of the subsets of our set X. Note that $\{a, b, c\}$ has 3 elements, and $\mathcal{P}(\{a, b, c\})$ has $2^3 = 2 \times 2 \times 2 = 8$ elements. So in this case, in fact in every case, $X \prec \mathcal{P}(X)$. Now let us arrange the 8 elements of $\mathcal{P}(\{a, b, c\})$ according to the subset relations that obtain among them, with the largest set, $\{a, b, c\}$, at the top of our diagram (called a *Hasse diagram*) and the smallest one, $\emptyset$, at the bottom. See Figure 2.

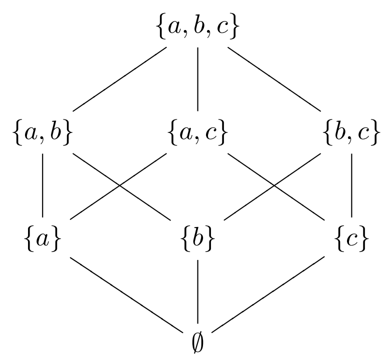


FIGURE 2 The Hasse diagram of $\mathcal{P}(\{a, b, c\})$

We have used the lines (*edges*) between set symbols here to indicate certain subset relations that obtain between the sets pictured in the diagram. Specifically, if you can move *up* from a set A to a set B along lines then $A \subseteq B$. Note that we have not for example drawn a line directly from $\{a\}$ to $\{a, b, c\}$. Our diagram allows us to infer that $\{a\} \subseteq \{a, b, c\}$, since it shows that $\{a\} \subseteq \{a, b\}$, and also $\{a, b\} \subseteq \{a, b, c\}$, and we know that subset is a *transitive* relation:

(1) Transitivity of subset: For all sets A, B, and C: if $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

Proof. Let A, B, and C be arbitrary sets. Assume that $A \subseteq B$ and $B \subseteq C$. We must show that $A \subseteq C$, that is, that an arbitrary element of A is also an element of C. Let x be an arbitrary element of A. Then x lies in B, since our first assumption says that everything in A is in B. But since x lies in B, we infer it also lies in C, since our second assumption says that everything in B is also in C. Thus given our assumptions, an arbitrary element of A is an element of C. So $A \subseteq C$, as was to be shown. □

There is one instance of the subset relation not depicted in Figure 2. Namely, we have not drawn a line from each set to itself to show that each set is a subset of itself. This is because we know that the subset relation is *reflexive*, that is, $Z \subseteq Z$, no matter what set Z we pick (even $Z = \emptyset$), as we have already seen. So here we rely on our general knowledge about sets to interpret the Hasse diagram. For the record,

(2) Reflexivity of subset: For all sets A, $A \subseteq A$.

(3) Non-subset-hood: $A \nsubseteq B$ iff there is some $x \in A$ which is not in B.

Exercise 2.1. Exhibit the Hasse diagram for each of:

a. $\mathcal{P}(\{a, b\})$.
b. $\mathcal{P}(\{a\})$.
c. $\mathcal{P}(\{a, b, c, d)\}$.
d. $\mathcal{P}(\emptyset)$.

Hasse diagrams of power sets incorporate more structure than meets the eye. Specifically they are *closed* under *intersection*, *union*, and *relative complement*:

Definition 2.1. Given sets A and B,

a. $A \cap B$ (read "A intersect B") is that set whose members are just the objects which are elements of both A and of B. For example,
   i. $\{a, b\} \cap \{b, c\} = \{b\}$,
   ii. $\{a, b\} \cap \{a, b, c\} = \{a, b\}$,
   iii. $\{a, b\} \cap .\{c\} = \emptyset$.
b. $A \cup B$ (read "A union B") is that set whose members are just the objects which are members of A or members of B (and possibly both). For example,
   i. $\{a, b\} \cup \{b, c\} = \{a, b, c\}$,
   ii. $\{b\} \cup \{a, b\} = \{a, b\}$,
   iii. $\{c, b\} \cup \emptyset = \{c, b\}$.
c. $A - B$ (read "A minus B") is the set whose members are those which are members of A but not of B. For example,
   i. $\{a, b, c\} - \{a, c\} = \{b\}$,
   ii. $\{a, c\} - \{a, b, c\} = \emptyset$,
   iii. $\{b, c\} - \emptyset = \{b, c\}$.

$A - B$ is also called *the complement of B relative to A*. Now to say that $\mathcal{P}(\{a, b, c\})$ is *closed under intersection*, $\cap$, is just to say that whenever A and B are elements of $\mathcal{P}(\{a, b, c\})$, then so is $A \cap B$. In fact, for all sets X, $\mathcal{P}(X)$ is closed under $\cap$.

Exercise 2.2. Let K be any collection of sets. What does it mean to say that K is closed under union? under relative complement? Is it true that for any set X, $\mathcal{P}(X)$ is closed under union and relative complement?

Exercise 2.3. Complete the following equations:

a. $EVEN \cap ODD =$ ________.
b. $EVEN \cup ODD =$ ________.
c. $\mathbb{N} - EVEN =$ ________.
d. $(\mathbb{N} - EVEN) \cap ODD =$ ________.
e. $(\mathbb{N} \cap EVEN) \cap ODD =$ ________.
f. $(\mathbb{N} \cap EVEN) \cap \{1, 3\} =$ ________.
g. $\{1, 2, 3\} \cap EVEN =$ ________.
h. $\{1, 2, 3, 4\} \cap ODD =$ ________.
i. $\{1, 2, 3\} \cap (EVEN \cup ODD) =$ ________.
j. $(\mathbb{N} \cap \emptyset) \cup \{0\} =$ ________.
k. $(ODD \cup ODD) \cap ODD =$ ________.

Exercise 2.4. Prove each of the statements below on the pattern used in (1). Each of these statements will be generalized later when we discuss the semantic interpretation of boolean categories.

a. For all sets A and B, $A \cap B \subseteq A$ (and also $A \cap B \subseteq B$).
b. For all sets A and B, $A \subseteq A \cup B$ (and also $B \subseteq A \cup B$).

Exercise 2.5. Some Boolean Truths of Set Theory

a. *Tests for subsethood.* Prove each of the three statements below for arbitrary sets A, B:
   i. $A \subseteq B$ iff $A \cap B = A$
   ii. $A \subseteq B$ iff $A - B = \emptyset$
   iii. $A \subseteq B$ iff $A \cup B = B$
b. *Idempotent laws.* For all sets A:
   i. $A \cap A = A$
   ii. $A \cup A = A$
c. *Distributivity laws.* For all sets A, B, C:
   i. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
   ii. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
d. *DeMorgan laws.* Writing $\neg A$ for $E - A$, where E is the domain of discourse:
   i. $\neg(A \cap B) = (\neg A \cup \neg B)$
   ii. $\neg(A \cup B) = (\neg A \cap \neg B)$

The exercises above basically assume that to define a set it is sufficient to say what its members are. Formally, the sufficiency is guaranteed by one of the axioms of the mathematical theory of sets. This is the Axiom of Extensionality; we have already met it on page 3. Once again, it says that for sets A and B,

$$A = B \text{ iff for all objects } x,\ x \in A \text{ iff } x \in B.$$

From this, it follows that if A and B are *different* sets, then one of them has a member that the other doesn't. Also, to prove that given sets A and B are the same (despite possibly having very different definitions), it suffices to show that each is a subset of the other. In fact, we have

(4) For all sets A, B, $A = B$ iff $A \subseteq B$ and $B \subseteq A$.

This is sometimes called the *anti-symmetric property of the inclusion relation* $\subseteq$.

2.1.1 Cardinality

We will be concerned at several points with the notion of the *cardinality* of a set. The idea here is that the cardinality of a set is a number which measures how big the set is. This "idea" is practically the definition in the case of finite sets, but to deal with infinite cardinalities one has to do a lot more work. We will not need infinite cardinals in this book at many places, so we only give the definition in the finite case and the case of a *countably infinite set*.

Definition 2.2. Let S be a finite set. If $S = \emptyset$, then the cardinality of S is 0, and we write $|S| = 0$. If $S \neq \emptyset$ then let k be the unique natural number such that $S \approx \{1, 2, \ldots, k\}$. Then the cardinality of S is k, and we write $|S| = k$. And when S is infinite, if $S \approx \mathbb{N}$, the set of natural numbers, then we say that S is *countably infinite* (or *denumerable*), and write, standardly, $|S| = \aleph_0$ (read "aleph zero"). $\aleph$ is the first letter of the Hebrew alphabet. This cardinal is also noted ω_0 (read "omega zero"), ω being the last letter in the Greek alphabet.

Here are some (largely familiar) examples of this notation: $|\emptyset| = 0$. For a any object $|\{a\}| = 1$. Similarly $|\{a, a\}| = 1$. For for a and b distinct objects, $|\{a, b\}| = 2$. And of course $|EVEN| = \aleph_0 = |\mathbb{N}|$. We should note that there are infinite sets that are larger than $\mathbb{N}$. We have already noted (and we prove shortly) that $X \prec \mathcal{P}(X)$, where X is any set and recall that $\mathcal{P}(X)$ is the set whose members are the subsets of X. So as a special case $\mathbb{N} \prec \mathcal{P}(\mathbb{N})$. However larger sets than $\mathbb{N}$ do not arise naturally in linguistic study.

Exercise 2.6.

a. Say why $|\{5, 5, 5\}| = |\{5\}|$.
b. Is $|\{5, (3 + 2)\}| = 2$? If not say why not.
c. Prove that for all sets A, B, if $A \subseteq B$ then $|A| \leq |B|$.
d. Exhibit a counterexample to the claim that if $A \subset B$ then $|A| < |B|$.
e. As a corollary to part b. above say why $|A \cap B| \leq |A|$ and $|A| \leq |A \cup B|$ for A, B any sets. (A *corollary* to a theorem P is one that follows from P in a fairly obvious way.)
f. Let the set *ODD* of odd natural numbers be given by the pseudolist $\{1, 3, 5, \ldots\}$. Show that $EVEN \approx ODD$.

2.1.2 Notation for Defining Sets

We have already seen how to define sets by *listing*: write down names of the elements of the set separated by commas and enclose the list in curly brackets. For example $\{0, 1\}$ is a 2 element set. But definition by listing is inherently limited. You can't write down an infinite list, and even when you know the set to be defined is finite, to define it by listing each element of the set must have a name in whatever language you are writing in. In mathematical discourse this condition often fails. To be sure, in the language of arithmetic the natural numbers have names, namely '0', '1', '2', etc. But in the language of Euclidean geometry the points on the plane don't have proper names. Moreover when the set is large defining it by listing is often impractical. The set of social security numbers of American citizens could be listed but writing down the list is a poor use of our time.

We have also seen productive means of defining new sets in terms of old ones—ones that are already defined. So if we know what sets A and B are then we can refer to $A \cap B$, $A - B$, $A \cup B$, and $\mathcal{P}(A)$.

A last and perhaps most widely used way of defining new sets in terms of old is called (unhelpfully) definition by *abstraction*. Here, given a set A, we may define a subset of A by considering the set of those elements of A which meet any condition of interest. For example, we might define the set of even natural numbers by saying:

$$EVEN =_{df} \{n \in \mathbb{N} \mid \text{for some } m \in \mathbb{N}, n = 2m\}.$$

We read the right hand side of this definition as "the set of n in $\mathbb{N}$ which are such that for some m in $\mathbb{N}$, n is 2 times m". The definition assumes that $\mathbb{N}$ is already defined (it also assumes that multiplication is defined). More generally if A is any set and φ any formula, $\{a \in A \mid \varphi\}$ is the set of elements a in A of which φ is true. So note that, trivially, $\{a \in A \mid \varphi\}$ is always a subset of A. φ can be any formula, even a contradictory one. For example, consider

$$\{n \in \mathbb{N} \mid n < 5 \text{ and } n > 5\}.$$

Clearly no $n \in \mathbb{N}$ satisfies the condition of being both less than 5 and greater than 5, so the set defined above is just the empty set $\emptyset$.

Returning to the definition of *EVEN* above, an even more succinct way of stating the definition is given by:

$$EVEN =_{df} \{2n \in \mathbb{N} \mid n \in \mathbb{N}\}$$

Informally this commonly used format says: "Run through the numbers n in $\mathbb{N}$, for each one form the number $2n$, and consider the collection of numbers so formed. That collection is the set of even (natural) numbers."

2.2 Sequences

We are representing expressions in English (and language in general) as sequences of words, and we shall represent languages as sets of these sequences. Here we present some basic mathematical notation concerning sequences, notation that we use throughout this book.

Think of a sequence as a way of choosing elements from a set. A sequence of such elements is different from a set in that we keep track of the order in which the elements are chosen. And we are allowed to choose the same element many times. The number of choices we make is called the length of the sequence. For linguistic purposes we need only consider finite sequences (ones whose length is a natural number).

In list notation we denote a sequence by writing down names of the elements (or *coordinates* as they are called) of the sequence, separating them by commas, as with the list notation for sets, and enclosing the list in angled or round brackets, but never curly ones. By convention the first coordinate of the sequence is written leftmost, then comes the second coordinate, etc. For example, $\langle 2, 5, 2\rangle$ is that sequence of length three whose first coordinate is the number two, whose second is five, and whose third is two. Note that the sequence $\langle 2, 5, 2\rangle$ has three coordinates whereas the set $\{2, 5, 2\}$ has just two members.

A sequence of length 4 is called a 4-tuple; one of length 5 a 5-tuple, and in general a sequence of length n is called an n-tuple, though we usually say pair or ordered pair for sequences of length 2 and (ordered) triple for sequences of length 3.

If s is an n-ary sequence (an n-tuple) and i is a number between 1 and n inclusive (that is, $1 \leq i \leq n$) then we write s_i for the ith coordinate of s. Thus $\langle 2, 5, 2\rangle_1 = 2$, $\langle 2, 5, 2\rangle_2 = 5$, etc[1]. If s is a sequence of length

[1]In more technical literature we start counting coordinates at 0. So the first

n then $s = \langle s_1, \ldots, s_n \rangle$. The *length* of a sequence s is noted $|s|$. So $|\langle 2,5,2\rangle| = 3$, $|\langle 2,5\rangle| = 2$, and $|\langle 2\rangle| = 1$. The following is fundamental:

(5) a. To define a sequence s it is necessary, and sufficient, to (i) give the length $|s|$ of s, and (ii) say for each i, $1 \leq i \leq |s|$, what object s_i is.

b. Sequences s and t are identical iff $|s| = |t|$ and for all i such that $1 \leq i \leq |s|$, $s_i = t_i$

For example, the statements in (6a,b,c,d) are all proper definitions of sequences:

(6) a. s is that sequence of length 3 whose first coordinate is the letter c, whose second is the letter a, and whose third is the letter t. In list notation $s = \langle c, a, t\rangle$.

b. t is that sequence of length 4 given by: $t_1 = 5$, $t_2 = 3$, $t_3 = t_2$, and $t_4 = t_1$.

c. u is that sequence of length 7 such that for all $1 \leq i \leq 7$,

$$u_i \quad = \quad \begin{cases} 3 & \text{if } i \text{ is odd} \\ 5 & \text{if } i \text{ is even} \end{cases}$$

d. v is that sequence of length 3 whose first coordinate is the word *Mary*, whose second is the word *criticized*, and whose third is the word *Bill*.

We frequently have occasion to consider sets of sequences. The following notation is standard:

Definition 2.3. For A and B sets,

a. $A \times B$ is the set of sequences s of length two such that $s_1 \in A$ and $s_2 \in B$. We write

$$A \times B \quad =_{df} \quad \{\langle x, y\rangle | x \in A \text{ and } y \in B\}.$$

$A \times B$ is read "A cross B" and called the *Cartesian product of A with B*. Generalizing,

b. If $A_1, \ldots, A_k$ are sets then $A_1 \times \cdots \times A_k$ is the set of sequences s of length k such that for each i, $1 \leq i \leq k$, $s_i \in A_i$. We abbreviate $A \times A$ as A^2 and $A \times \cdots \times A$ (n times) as A^n. $A^0 = \{e\}$, where e is the unique (see below) sequence of length zero. A^* is the set of finite sequences of elements of A. That is, $s \in A^*$ iff for some natural number n, $s \in A^n$.

$|A \times B|$, the cardinality of the set $A \times B$, is exactly the product $|A| \times |B|$. This is what accounts for the notation. We have $|A|$ many

---

coordinate of an n-tuple s would be noted s_0 and its nth would be noted s_{n-1}.

choices for the first element of a pair in $A \times B$ and $|B|$ many choices for the second. Thus we have $|A| \times |B|$ choices in toto. So $|A \times A| = |A|^2$, and more generally $|A^n| = |A|^n$.

Exercise 2.7.

a. Exhibit the sequences (6b,c,d) in list notation.
b. Answer the following True or False; for a false statement explain why it is false.
   a. $\langle 2, 4, 6\rangle_2 = 2$.
   b. $|\langle 3\rangle| > 1$.
   c. $|\langle 3, 3, 3\rangle| = 3$.
   d. for some i between 1 and 3 inclusive, $\langle c, a, t\rangle_i = b$.
   e. for some $i < j$ between 1 and 3 inclusive, $\langle c, a, t\rangle_i = \langle c, a, t\rangle_j$.
c. Let $A = \{a, b, c\}$ and $B = \{1, 2\}$. Exhibit the following sets in list notation:
   i. $A \times B$.
   ii. $B \times A$.
   iii. $B \times B$.
   iv. $B \times (A \times B)$.

Note that a sequence of length zero has no coordinates. And from (5b) there cannot be two different sequences both of length zero since they have the same length and do not differ at any coordinate. Moreover,

(7) There is a sequence of length zero, called the empty sequence, often noted e.

One widely used binary operation on sequences is concatenation, noted $\frown$.

Definition 2.4. If s is a sequence $\langle s_1, \ldots, s_n\rangle$ of length n and t a sequence $\langle t_1, \ldots, t_m\rangle$ of length m then $s \frown t$ is that sequence of length $n + m$ whose first n coordinates are those of s and whose next m coordinates are those of t. That is, $s \frown t =_{df} \langle s_1, \ldots, s_n, t_1, \ldots, t_m\rangle$.

For example, $\langle 3, 2\rangle \frown \langle 5, 4, 3\rangle = \langle 3, 2, 5, 4, 3\rangle$. Similarly, we have that $\langle 1\rangle \frown \langle 1\rangle = \langle 1, 1\rangle$.

Observe that concatenation is *associative*:

(8) $(s \frown t) \frown u = s \frown (t \frown u)$.

For example, (9a) = (9b):

(9) a. $(\langle 3, 4\rangle \frown \langle 5, 6, 7\rangle) \frown \langle 8, 9\rangle = \langle 3, 4, 5, 6, 7\rangle \frown \langle 8, 9\rangle = \langle 3, 4, 5, 6, 7, 8, 9\rangle$

b. $\langle 3,4\rangle \frown (\langle 5,6,7\rangle \frown \langle 8,9\rangle) = \langle 3,4\rangle \frown \langle 5,6,7,8,9\rangle =$
$\langle 3,4,5,6,7,8,9\rangle$

So as with intersection, union, addition, etc. we omit parentheses and write simply $s \frown t \frown u$. But note that in distinction to $\cap$ and $\cup$, concatenation is not commutative:

$$\langle 0\rangle \frown \langle 1\rangle = \langle 0,1\rangle \neq \langle 1,0\rangle = \langle 1\rangle \frown \langle 0\rangle.$$

The empty sequence e exhibits a distinctive behavior with respect to concatenation. Since e adds no coordinates to anything it is concatenated with we have:

(10) For all sequences s, $s \frown e = e \frown s = s$.

The role that e plays with regard to $\frown$ is like that which 0 plays with regard to addition ($n + 0 = 0 + n = n$) and the $\emptyset$ plays with regard to union ($A \cup \emptyset = \emptyset \cup A = A$). e is called an *identity element* with respect to concatenation, just as 0 is an identity element with respect to addition and $\emptyset$ an identity element with respect to union.

Note that just as sets can be elements of other sets, so sequences can be coordinates of other sequences. For example the sequence $s = \langle 4, \langle 3,5,8\rangle\rangle$ is a sequence of length 2. Its first coordinate is the number 4, is second a sequence of length 3: $\langle 3,5,8\rangle$. That is, $s_1 = 4$ and $s_2 = \langle 3,5,8\rangle$. Observe:

(11) a. $|\langle j,o,h,n,c,r,i,e,d\rangle| = 9$
b. $|\langle\langle j,o,h,n\rangle, \langle c,r,i,e,d\rangle\rangle| = 2$

(11a) is a 9-tuple of English letters. (11b) is a sequence of length 2, each of whose coordinates is a sequence of letters. If we call these latter sequences words then (11b) is a two coordinate sequence of words, that is, an (ordered) pair of words.

Exercise 2.8. Answer True or False to each statement below. If False, say why.

a. $|\langle c,a,t\rangle| < |\langle\langle e,v,e,r,y\rangle, \langle c,a,t\rangle\rangle|$.
b. $|\langle a,b,a\rangle| = |\langle b,a\rangle|$.
c. $|\langle 0,0,0\rangle| < |\langle 1000\rangle|$.
d. $|\langle 2,3,4\rangle \frown e| > |\langle 1,1,1\rangle|$.
e. for all finite sequences s, t, $s \frown t = t \frown s$.
f. $\langle 2+1, 32\rangle = \langle 3, 23\rangle$.
g. For all finite sequences s, t,
   i. $|s \frown t| = |s| + |t|$; and
   ii. $|s \frown t| = |t \frown s|$.

Exercise 2.9. Compute stepwise the concatenations in (a) and (b) below, observing that they yield the same result, as predicted by the associativity of concatenation.

a. $(\langle a, b\rangle \frown \langle b, c, d\rangle) \frown \langle b, a\rangle =$
b. $\langle a, b\rangle \frown (\langle b, c, d\rangle \frown \langle b, a\rangle) =$

Exercise 2.10. Fill in the blanks below, taking s to be $\langle 0, \langle 0, 1\rangle, \langle 0, \langle 0, 1\rangle\rangle\rangle$.

a. $s_1 =$ ________
b. $s_2 =$ ________
c. $(s_2)_2 =$ ________
d. $s_3 =$ ________
e. $(s_3)_2 =$ ________
f. $((s_3)_2)_1 =$ ________

Prefixes and subsequences Given a sequence s, a *prefix* of s is a piece of s that comes "at the front." For example, if $s = \langle 2, 4, 5, 1\rangle$, then the prefixes of s are ϵ (the empty string), $\langle 2\rangle$, $\langle 2, 4\rangle$, $\langle 2, 4, 5\rangle$, and s itself.

Similarly, a *subsequence* of s is a string t that comes "somewhere inside" s. That is, t might not be at the very front or the very end. For example, the substrings of s are the prefixes of s listed above, and also $\langle 4\rangle$, $\langle 4, 5\rangle$, $\langle 4, 5, 1\rangle$, $\langle 5\rangle$, $\langle 5, 1\rangle$, and $\langle 1\rangle$. But we would not count a string like $\langle 2, 5\rangle$ as a substring of s; we want substrings to be *connected occurrences*.

Exercise 2.11. Let V be a non-empty set and fix a string $s \in V^*$ (V^* recall is the set of finite sequences of elements of V). Then consider the following sets:

a. $\{t : \text{for some } w \in V^*, s = t \frown w\}$.
b. $\{t : \text{for some } w \in V^*, s = w \frown t\}$.
c. $\{t : \text{for some } w \in V^*, \text{ for some } u \in V^*, s = w \frown t \frown u\}$.

Which of these defines the set of substrings of s? Which defines the set of prefixes of s? What would should we call the remaining set?

2.3 Functions and Sequences

We have seen that a function f from a set A into a set B associates with each $\alpha \in A$ a unique object noted $f(\alpha)$ in B. $f(\alpha)$, recall, is called the value of the function f at the argument α. If A was a small set whose members had names we could define f just by listing the objects α in A and next to each we note the object in B that f maps α to, namely $f(\alpha)$. Suppose for example that A is the set $\{1, 2, 3\}$ and B is the set $\{1, 4, 9\}$ and that $f(1) = 1$, $f(2) = 4$, and $f(3) = 9$. Then we could define this f just by listing pairs (sequences of length 2).

(12) $f = \{\langle 1, 1\rangle, \langle 2, 4\rangle, \langle 3, 9\rangle\}$

In this way we think of a function f from A into B as a particular subset of $A \times B$, namely $f = \{\langle \alpha, f(\alpha)\rangle \mid \alpha \in A\}$. So we represent f as a set of sequences of length 2. This set of pairs is called the *graph* of f. Consider the squaring function SQ from $\mathbb{N}$ to $\mathbb{N}$. $SQ(n) =_{df} n \cdot n$, all $n \in \mathbb{N}$. (Usually $SQ(n)$ is noted n^2). The domain, $\mathbb{N}$, of SQ is infinite, and the graph of SQ is given in (13).

(13) $SQ = \{\langle n, SQ(n)\rangle \mid n \in \mathbb{N}\}$

We could even *define* a function F from a set A into a set B as a subset of $A \times B$ meeting the condition that for every $\alpha \in A$ there is exactly one $\beta \in B$ such that $\langle \alpha, \beta\rangle \in F$. As long as F meets this condition then we can define $F(\alpha)$ to be the unique β such that $\langle \alpha, \beta\rangle \in F$. This is what it means to say that β is given as a function (F) of α.

Some terminology. A two-place function g from A into B is a function from $A \times A$ into B. So such a g maps each pair $\langle \alpha, \alpha'\rangle$ of objects from A to a unique element of B. More generally an n-place function from A into B is a function from A^n into B. We use the phraseology "n-place function on A" to refer to a function from A^n into A. For example the addition function is a 2-place function on $\mathbb{N}$. It maps each pair $\langle n, m\rangle$ of natural numbers to the number $(n+m)$. Notice here that in writing traditional expressions we tend to write the function symbol, '+' in this case, between its two arguments.

The illustrative functions on numbers utilized so far used are defined in terms of addition, multiplication, and exponentiation. But several of the ones we used earlier to describe syntactic structure were defined in a non-trivial *recursive* way. Namely the values at small arguments were given explicitly (by listing) and then their values at larger arguments were defined in terms of their values at lesser arguments. Recall for example the function F from $\mathbb{N}$ into the set of English expressions as follows:

(14) a. $F(0) =$ *the President* and for all $n > 0$,
b. $F(n) = \langle \textit{the} \frown \textit{mother} \frown \textit{of} \frown F(n-1)\rangle$

So we compute that $F(2) =$ *the mother of* $F(1) =$ *the mother of the mother of* $F(0) =$ *the mother of the mother of the President*. Recursive functions of this sort are widely used in computer science. Here are two household examples, the *factorial* function **Fact** and the *Fibonacci* function **Fib**, both from $\mathbb{N}$ into $\mathbb{N}$:

(15) a. $\mathbf{Fact}(0) = 1$ and for all $n > 0$, $\mathbf{Fact}(n) = n{\cdot}\mathbf{Fact}(n-1)$ ($\mathbf{Fact}(n)$ is usually denoted $n!$).
b. $\mathbf{Fib}(0) = 1$, $\mathbf{Fib}(1) = 1$, and for all $n \in \mathbb{N}$, $\mathbf{Fib}(n+2) = \mathbf{Fib}(n+1){+}\mathbf{Fib}(n)$.

In defining the Fibonacci sequence **Fib** above we gave its value explicitly at both 0 and 1, and then defined later values in terms of the two previous values. You might use a similar (not identical) tack in the following exercise:

Exercise 2.12. Let K be the following set of informally given expressions:
$K = \{$a whisky lover, a whisky lover hater,
a whisky lover hater lover, a whiskey lover hater lover hater, $\ldots\}$
Note that the "-er" words at the end of each expression alternate between 'lover' and 'hater'. For example, 'a whiskey lover lover' is not in the set, nor is 'a whiskey lover hater', etc. Define a one to one function from $\mathbb{N}$ onto K.

Bijections. When we define a function F from some A into some B we specify B as the codomain of F. The *range* of F, $\mathrm{Ran}(F)$, is by definition the set of values F can take, that is, $\{F(\alpha) \mid \alpha \in A\}$. If $\mathrm{Ran}(F) = B$ then F is said to be *onto* B and is called *surjective* (or a *surjection*). If in addition F is one-to-one (injective, an injection) then F is *bijective* (a *bijection*). A *bijection* from a set A to itself is called a *permutation* of A. In general bijections play a fundamental role in defining what we mean by *sameness* of structure. But for the moment we just consider two examples.

(16) Let H be that function from $\{a, b, c, d\}$ into itself given by the table below:

$$\begin{array}{r|cccc} x & a & b & c & d \\ H(x) & b & c & d & a \end{array}$$

So $H(a) = b$, $H(b) = c$, $H(c) = d$, and $H(d) = a$. Clearly H is both one-to-one and onto, so H is a permutation of $\{a, b, c, d\}$. Suppose we think of a as the upper left hand corner of a square, b as the lower left hand corner, c as the lower right hand corner, and d as the upper right hand corner. Then H would represent a 90 degree rotation counter-clockwise.

As a second example define a function $\neg$ from $\mathcal{P}(\{a, b, c\})$ to itself as follows: for all subsets A of $\{a, b, x\}$, $\neg(A) = \{a, b, c\} - A$.

Exercise 2.13. Exhibit the table of $\neg$ defined above. (So you must list the subsets of $\{a, b, c\}$—there are 8 of them—and next to each give its value under $\neg$.)

We will have much more to say about bijections in later chapters, but first we generalize the set theoretic operations of intersection and union in useful ways.

2.4 Arbitrary Unions and Intersections

Suppose we have a bunch of sets (we don't care how many). Then the *union* of the bunch is just that set whose members are the things in at least one of the sets in the bunch. And the *intersection* of the bunch is the set of objects that lie in each set in the bunch (if no object meets that condition then the intersection of the bunch is just the empty set). Let us state this now a little more carefully.

Given a domain (or universe) E of objects under study, let K be any set whose members are subsets of E. In such a case we call K a *family* (or *collection*) of subsets of E. Then we define:

(17) a. $\bigcup K =_{df} \{x \in E \mid \text{for some } A \in K, x \in A\}$
b. $\bigcap K =_{df} \{x \in E \mid \text{for all } A \in K, x \in A\}$

$\bigcup K$ is read "union K" and $\bigcap K$ is read "intersect K".

Here is an example. Let E be $\mathbb{N}$. Let K be the collection of subsets of $\mathbb{N}$ which have 5 as a member. Then $\{5\} \in K$, $\mathit{ODD} \in K$, $\mathbb{N} \in K$, and $\{3,5,7\} \in K$. But $\emptyset \notin K$, nor are EVEN or $\{n \in \mathbb{N} \mid n > 7\}$.

If K is a finite collection of sets, say $K = \{A_1, A_2, A_3, A_4, A_5\}$, then $\bigcup K$ might be noted with a notation such as $\bigcup_{1 \le i \le 5} A_i$. We would even be likely to write simply $A_1 \cup \cdots \cup A_5$. In cases such as this the set K is called an *indexed* family of sets. In this example the index set is $\{1,2,3,4,5\}$. We often use letters like I and J for arbitrary index sets, and when say "Let K be an indexed family of sets" we mean that for some set I, $K = \{A_i \mid i \in I\}$. And now arbitrary unions and intersections are noted $\bigcup_{i \in I} A_i$ and $\bigcap_{i \in I} A_i$. Writing out their definitions explicitly we have:

(18) a. $\bigcup_{i \in I} A_i =_{df} \{x \in E \mid \text{for some } i \in I, x \in A_i\}$
b. $\bigcap_{i \in I} A_i =_{df} \{x \in E \mid \text{for all } i \in I, x \in A_i\}$

When the index set I is clear from context (or unimportant) we just write $\bigcup_i A_i$ and $\bigcap_i A_i$.

2.5 Definitions by Closure (Recursion)

Many of our definitions of linguistic sets will use this format. There are several different ways of formulating this type of definition in the literature. The one we opt for here is not the most succinct but is the best for understanding the core idea behind the definitional technique.

Definition 2.5. Given a set A, a function f from A into A, and a subset K of A, we define the *closure of* K *under* f, noted $\mathrm{Cl}_f(K)$, as follows:

a. Set $K_0 = K$ and for all $n \in \mathbb{N}$, $K_{n+1} = K_n \bigcup \{f(x) \mid x \in K_n\}$.
b. $\mathrm{Cl}_f(K) =_{df} \{x \in A \mid \text{for some } n,\ x \in K_n\}$. That is, $\mathrm{Cl}_f(K) = \bigcup_n K_n$.

So the idea is that we start with some given set K and add in all the things we can get by applying f to elements of K. That gives us K_1. Then repeat this process, obtaining K_2, and then continue ad infinitum. The result is all the things you can get starting with the elements of K and applying the function any finite number of times. In the linguistic cases of interest K will be a lexicon for a language and f (actually there may be several f) will be the structure building functions that start with lexical items and construct more complex expressions. Here are two simple examples.

Example 1. Let $V = \{a, b\}$, f a map from V^* into V^* defined by: $f(s) = a \frown s \frown b$, and $K = \{ab\}$. (Here we write simply ab for the sequence $\langle a, b\rangle$; recall that for sequences of symbols we often omit the commas and angled brackets). Then $K_0 = \{ab\}$, $K_1 = \{ab, aabb\}$, $K_2 = \{ab, aabb, aaabbb\}$, etc. Then provably $\mathrm{Cl}_f(\{ab\}) = \{a^n b^n \mid n > 0\}$.

Example 2. Let $V = \{a, b\}$, and let Copy be a function from V^* to V^* given by $\mathrm{Copy}(s) = s \frown s$, and $K = \{b\}$. Then $K_0 = \{b\}$, $K_1 = \{b, bb\}$, $K_2 = \{b, bb, bbbb\}$, etc., and $\mathrm{Cl}_{\mathrm{Copy}}(\{b\}) = \{b^2 \mid n \in \mathbb{N}\}$.

A fundamental theorem concerning definitions by closure is given in Theorem 2.1.

Theorem 2.1. *For all functions f and sets K, $Cl_f(K)$ has the following three properties:*

a. $K \subseteq Cl_f(K)$,
b. $Cl_f(K)$ *is closed under f. That is, whenever $x \in Cl_f(K)$ then $f(x) \in Cl_f(K)$, and*
c. if M is any set which includes K and is closed under f then $Cl_f(K) \subseteq M$. (This is what we mean when we call $Cl_f(K)$ the least *set that includes K and is closed under f.)*

Fact. $\mathbb{N}$ is the closure of $\{0\}$ under $+1$, the addition of 1 function. So if M is a set with $0 \in M$ and M is closed under addition of 1 then M contains all the natural numbers. So to show that all numbers have some property φ we just let M be the set of numbers with φ and show that $0 \in M$ and M is closed under $+1$. This is usually called the *Principal of Induction*.

Definition by Closure as given above generalizes in several largely obvious ways. It makes obvious sense to form the closure of a set under several functions, not just one, and the functions need not be unary. We may naturally speak of the closure of a set of numbers under multiplication for example. An only slightly less obvious generalization concerns closure under partial functions.

Definition 2.6. A *partial function* f from A to B is a function whose domain is a subset of A and whose codomain is included in B.

For example, a function whose domain was the set of even numbers which mapped each one to its square would be a partial function from $\mathbb{N}$ to $\mathbb{N}$.

In defining closure under partial functions we need a slight addition to the definition of K_{n+1} as follows: $K_{n+1} = K \bigcup \{f(x) \mid x \in K \text{ and } x \in \text{Dom}(f)\}$.

Exercise 2.14.

a. Let Sq be that function from $\mathbb{N}$ to $\mathbb{N}$ given by: $\text{Sq}(n) = n \cdot n$.
   i. Set $K = \{2\}$. Exhibit K_1, K_2, and K_3. Say in words what $\text{Cl}_{\text{Sq}}(\{2\})$ is.
   ii. Do the same with $K = \{1\}$.
   iii. Do the same with $K = \emptyset$. Is $\emptyset$ closed under Sq?
   iv. What is $\text{Cl}_{\text{Sq}}(\mathit{EVEN})$.

b. Consider the set $V = \{P, Q, R, \&, \text{NEG},), (, \}$.
   i. Define a unary function NEG from $V^* \to V^*$ which prefixes 'NEG' to each $s \in V^*$.
   ii. Define a binary function AND which maps each pair $\langle s, t\rangle$ of elements to its conjunction.
   iii. Write AF ("atomic formula") for $\{P, Q, R\}$. So AF_0 is $\{P, Q, R\}$. Consider the closure of AF under NEG and AND.
      i. What is the least n such that NEG(P & NEGQ) is in AF_n?
      ii. Give an explicit argument that NEG(P & NEGQ) $\in \text{Cl}_{\text{AND,NEG}}(\text{AF})$. Your argument might begin: $Q \in \text{AF}_0$. So NEG$Q \in \text{AF}_1$...
      iii. Prove that for all $\varphi \in \text{Cl}_{\text{AND,NEG}}(\text{AF})$ the number of parenthesis occurring in φ is even. Here is how to set up the proof:
      Let $M = \{\varphi \in \text{Cl}_{\text{AND,NEG}}(\text{AF})$
      $\mid$ the number of parenthesis in φ is even$\}$.
      Step 1: Show that AF $\subseteq M$.

Step 2: Show that M is closed under NEG and AND.

Note that AND and NEG are functions defined on V^* and so apply to strings over that vocabulary. For example, NEG(&&))Q) = NEG$\frown$ &&))Q and AND(QNEG((, RRR) = (QNEG(($\frown$ & $\frown$). The functions must be defined independently of the set we form the closure of.

Exercise 2.15. Prove that for all sets A, $\bigcup \mathcal{P}(A) = A$.

We conclude this chapter by stating and proving a fundamental theorem in set theory due to Georg Cantor in the late 1800s. For the linguistic material covered in subsequent chapters of this book it is not necessary to understand the proof of this theorem. But it is one of the foundational theorems in set theory, and our brief introduction actually gives us (almost) enough to follow the proof, so we give it here. The basic claim of the theorem is that *any* set is strictly smaller than its power set. So forming power sets is one way of forming new sets from old that always leads to bigger sets. In presenting the properly formal version of the theorem we do need to assume the Schroeder-Bernstein mentioned earlier and stated more rigorously here (we give a proof at the end of Chapter 8):

Theorem 2.2 (Schroeder-Bernstein)**.** *For all sets A, B, $A \approx B$ iff there is a bijection from A into B.*

(Our original definition of $\approx$ just says there is an injection from A to B *and* an injection from b to A.)

Theorem 2.3 (Cantor)**.** *For all sets A, $A \prec \mathcal{P}(A)$.*

You might try a few simple examples to see that the theorem holds for them. For example, $\emptyset \prec \mathcal{P}(\emptyset) = \{\emptyset\}$ since $|\emptyset| = 0$ and $|\{\emptyset\}| = 1$. Similarly $\{a\} \prec \mathcal{P}(\{a\}) = \{\emptyset, \{a\}\}$, so $\{a\}$ has one element, $\mathcal{P}(\{a\})$ has 2. If A has just 2 elements then $\mathcal{P}(A)$ has 4 elements. If A has just 3 elements $\mathcal{P}(A)$ has 8. In general for A finite with n elements, how many elements does $\mathcal{P}(A)$ have?

We now prove theorem 2.3.

Proof. Let A be an arbitrary set. We show that $A \prec \mathcal{P}(A)$. First we show that $A \preceq \mathcal{P}(A)$. Clearly the function f mapping each element α in A to $\{\alpha\}$ is a map from A into $\mathcal{P}(A)$ which is one to one. Thus $A \preceq \mathcal{P}(A)$.

Now we show that there is no surjection from A to $\mathcal{P}(A)$. From this it follows that there is no bijection from A to $\mathcal{P}(A)$. By Schroeder-Bernstein then $A \not\approx B$. Thus $A \prec \mathcal{P}(A)$.

Let h be an arbitrary function from A into $\mathcal{P}(A)$. We show that h is not onto. For each $x \in A$, $h(x) \subseteq A$, so it makes sense to ask whether

$x \in h(x)$ or not. Let us write K for $\{x \in A \mid x \notin h(x)\}$. Trivially $K \subseteq A$, so $K \in \mathcal{P}(A)$. We show that for every $\alpha \in A$, $h(\alpha) \neq K$, whence we will infer that h is not onto. Suppose, leading to a contradiction, that there is an α in A such that $h(\alpha) = K$. Then either $\alpha \in K$ or $\alpha \notin K$ (this is a logical truth). If $\alpha \in K$ then since $\alpha \in A$ we infer by the defining condition for K that $\alpha \notin h(\alpha)$. But $h(\alpha)$ is K, so we have a contradiction: $\alpha \in K$ and $\alpha \notin K$. Thus the assumption that $\alpha \in K$ is false. So $\alpha \notin K$. But again since $h(\alpha) = K$ then $\alpha \notin h(\alpha)$ and since $\alpha \in A$, α satisfies the condition for being in $h(\alpha)$. And since $h(\alpha) = K$ we infer that $\alpha \in K$, contradicting that $\alpha \notin K$. Thus the assumption that there is an α such that $h(\alpha) = K$ is false. So for all α, $h(\alpha) \neq K$. Thus h is not onto. Since h was an arbitrary function from A into $\mathcal{P}(A)$ we have that all such h fail to be onto. Thus $A \not\approx B$, so $A \prec \mathcal{P}(A)$ completing the proof. □

Corollary 2.4. $\mathbb{N} \prec \mathcal{P}(\mathbb{N})$.

This follows immediately as a special case of Cantor's Theorem.

Now let us define the following infinite sequence H of sets:

(19) $H(0) =_{df} \mathbb{N}$ and for all n, $H(n+1) =_{df} \mathcal{P}(H(n))$.

Thus we have an infinite sequence of increasingly large infinite sets:

(20) $\mathbb{N} \prec \mathcal{P}(\mathbb{N}) \prec \mathcal{P}(\mathcal{P}(\mathbb{N})) \prec \cdots$.

2.6 Bijections and the Sizes of Cross Products

Cantor's power set theorem shows that forming power sets always leads to bigger sets, since $A \prec \mathcal{P}(A)$, all A. It is natural to wonder whether there are other operations that always lead to bigger sets. Here are two natural candidates (neither of which works, but this is not obvious without proof).

First if A and B are disjoint, non-empty finite sets then $A \prec A \cup B$ and $B \prec A \cup B$. So union looks like it leads to larger sets, but this fails when A or B are infinite. Consider first a near trivial case: Let $A = \{a\} \cup \mathbb{N} = \{a, 0, 1, \ldots\}$. Consider the mapping in (21b), indicated informally in (21a):

(21) a.
$$\begin{array}{rcllll} x & = & a & 0 & 1 & \cdots \\ F(x) & = & 0 & 1 & 2 & \ldots \end{array}$$

b. Define $F : A \to \mathbb{N}$ by setting $F(a) = 0$ and for all natural numbers n, $F(n) = n + 1$.

It is easy to see that F is a bijection from A into $\mathbb{N}$, thus $A \approx \mathbb{N}$ by the Schroeder-Bernstein Theorem. Still, we may wonder if the union of two disjoint infinite sets leads to sets bigger than either of those we take the union over. Again the answer is no. Here is simple, non-trivial, case:

(22) Set $\mathbb{N}^+ = \{1, 2, \ldots\}$ and $\mathbb{N}^- = \{-1, -2, \ldots\}$. Then $\mathbb{N} \approx \mathbb{N}^+ \cup \mathbb{N}^-$. Elements of $\mathbb{N}^+$ are called *positive integers* and elements of $\mathbb{N}^-$ are *negative integers*.

A bijection from $\mathbb{N}^+ \cup \mathbb{N}^-$ into $\mathbb{N}$ is informally sketched in (23a) and given formally in (23b):

(23) a.
$$\begin{array}{cccccccc} 1 & -1 & 2 & -2 & 3 & -3 & \cdots \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ 1 & 2 & 3 & 4 & 5 & 6 & \cdots \end{array}$$

b. Define F from $\mathbb{N}^+ \cup \mathbb{N}^-$ into $\mathbb{N}$ by:

$$F(n) = \begin{cases} 2n - 1 & \text{if } n \text{ is positive} \\ -2n & \text{if } n \text{ is negative.} \end{cases}$$

Clearly F is one to one, and is in fact onto. So $\mathbb{N}^+ \cup \mathbb{N}^- \approx \mathbb{N}$. Thus taking (finite) unions does not always increase size—indeed when A and B are infinite $A \cup B$ has the same size as the larger of A, B (that is, $A \cup B \approx A$ if $B \preceq A$, otherwise $A \cup B \approx B$).

Does forming cross products produce sets larger than the ones we form the product from? Certainly for A and B finite sets each with at least two elements $A \prec A \times B$ and $B \prec A \times B$ since $|A \times B| = |A| \cdot |B|$. But what about the infinite case? It doesn't seem silly to think that there are more pairs of natural numbers than natural numbers. After all, each natural number is paired with infinitely many others in forming $\mathbb{N} \times \mathbb{N}$, and there are infinitely many such sets of pairs in $\mathbb{N} \times \mathbb{N}$. But again it turns out that $\mathbb{N} \times \mathbb{N} \approx \mathbb{N}$, though it is harder to see than in the case of unions. (24a) gives an explicit bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$, and (24b) provides a pictorial representation illustrating the bijection (Büchi (1989) pg. 22, Enderton (1977) pg. 130).

(24) a. Define $F : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by
$F(x, y) = x + ((x + y + 1)(x + y)/2.$

b.

	0	1	2	3	4	5	$\cdots$
0	0	2	5	9	14	20	$\cdots$
1	1	4	8	13	19	$\ddots$	
2	3	7	12	18	$\ddots$		
3	6	11	17	$\ddots$			
4	10	16	$\ddots$				
5	15	$\ddots$					
$\vdots$	$\vdots$						

We close this section by noting three basic characteristics of bijections, ones that arise often in formal studies. First, for any set A, let us define a function Id_A from A to A by:

(25) For all $\alpha \in A$, $\mathrm{Id}_A(\alpha) = \alpha$.

Id_A is the *identity* function on A. It is clearly a bijection. Second, for A, B, C arbitrary sets,

(26) If h is a bijection from A to B then h^{-1} (read: "h inverse") is a bijection from B to A, where for all $\alpha \in A$, all $\beta \in B$,

$$h^{-1}(\beta) = \alpha \text{ iff } h(\alpha) = \beta.$$

In other words, h^{-1} maps each β in B to the unique α in A which h maps to β. (There is such an α because h is onto, and there isn't more than one because h is one to one). So h^{-1} just runs h backwards. The domain of h^{-1} is B, h^{-1} maps distinct β and β' to distinct α and α' in A (otherwise h would not even be a function) so h^{-1} is one to one, and finally h^{-1} is onto since each α in A gets mapped to some β in B. And thirdly,

(27) Given a bijection h from A to B and a bijection g from B to C we define a bijection noted $g \circ f$ (read: "g compose f") from A to C as follows: for all $\alpha \in A$,

$$(g \circ f)(\alpha) = g(f(\alpha)).$$

Clearly $g \circ f$ is one to one: if $\alpha \neq \alpha'$ then $f(\alpha) \neq f(\alpha')$ since f is one to one, thus $(g \circ f)(\alpha) = g(f(\alpha)) \neq g(f(\alpha'))$, since g is one to one, this equals $g \circ f(\alpha'))$. And since f is onto, the range of f is all of B, and since g is onto the range of $g \circ f$ is all of C, so $g \circ f$ is onto, whence $g \circ f$ is a bijection.

So far we have just used bijections to show that two sets are the same size. Deeper uses arise later in showing the different mathematical structures are isomorphic. Even here these definitions allow us to draw a few conclusions that reassure us that we have provided a reasonable characterization of "has the same size as". For one, if $A \approx B$, i.e. there is a bijection h from A into B, then h^{-1} is a bijection from B into A, so we infer that B has the same size as A. If our mathematical definition of *has the same size as* allowed that some A had the same size as some B but did not support the inference that B had the same size as A then our mathematical definition would have failed to capture our pretheoretical notion. Similarly the fact that the composition of two bijections is a bijection assures us that if A has the same size as B, and B the same size as C, then A has the same size as C.

Some notation and terminology.

1. As already noted, a set of the same size as $\mathbb{N}$ is said to be *denumerable*. A set which can be put in a one to one correspondence with a subset of $\mathbb{N}$ is called *countable*. So countable sets are either finite or denumerable. When a countable set is denumerable we tend to say it is *countably infinite*.
2. We often abbreviate the statement $x \in A$ and $y \in A$ by $x, y \in A$. And we abbreviate $A \subseteq X$ and $B \subseteq X$ by $A, B \subseteq X$.

2.7 Suggestions for Further Study

If you are interested in getting more background on set theory, we recommend Enderton (1977) and Halmos (1974).

2.8 Addendum: Russell's Paradox

In defining sets by abstraction we require that the set so defined be a subset of an antecedently given set. And we consider all the members of that set that satisfy whatever requirement we are interested in. This approach is one way to avoid what is known as *Russell's Paradox*, which we give here for the interested reader, noting that paradoxes tend to be confusing and understanding the paradox deeply is a not a prerequisite for following the rest of this book. The paradox, historically very important in the development of set theory, arises if we allow ourselves to (try to) define sets without requiring that the elements in the defined set be drawn from an antecedently given set. If we could do this then (this begins Russell's Paradox) we would consider A below to be a set:

(28) $A = \{x \mid x \text{ is a set and } x \notin x\}$.

So suppose that A, as apparently defined, is a set. Is it a member of itself? Suppose (leading to a contradiction) that $A \in A$. Then A fails the condition for being in A, so $A \notin A$, a contradiction. Thus the assumption that $A \in A$ is false. That is, $A \notin A$. But then, since A is a set by assumption, A meets the conditions for being a member of A. That is, $A \in A$, contradiction. Thus the assumption that A was a set must be false.

3

Syntax I: Trees and Order Relations

We think of a *language* as a set of meaningful, pronounceable (or signable) expressions. A *grammar* is a definition of a language. As linguists, we are interested in defining (and then studying) languages whose expressions are approximately those of one or another empirically given natural language (English, Japanese, Swahili, ...). If a proposed grammar of English, for example, failed to tell us that *Every cat chased every mouse* is an expression of English, that grammar would be *incomplete*. If it told us that *Cat mouse every every chased* is an expression of English, it would be *unsound*. So designing a sound and complete grammar for a natural language involves considerable empirical work, work that teaches us much about the structure and nature of human language.

And as we have seen in Chapter 1, a grammar for a natural language cannot just be a finite list of expressions: natural languages present too many expressions to be listed in any enlightening way. Moreover, a mere list fails to account for the *productivity* of natural language—our ability to form and interpret novel utterances—as it fails to tell us how the form and interpretation of complex expressions depends on those of its parts.

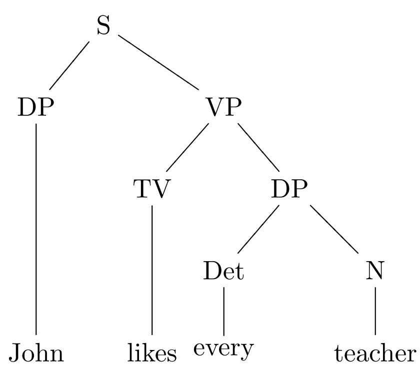
Consequently, a *grammar* G of a language is presented in two gross parts: (1) a *Lexicon*, that is, a finite list of expressions called *lexical items*, and (2) a set of *Rules* which iteratively derive complex expressions from simpler ones, beginning with the lexical items. The language $L(G)$ generated by the grammar is then the lexical items plus all those expressions constructable from them by applying the rules finitely many times.

In this chapter, we present some standard techniques and formalisms linguists use to show how complex expressions incorporate simpler ones. So here we concentrate on generative syntax, ignoring both semantic representation (see Chapters 7–9) and phonological representation. In Chapter 4, we consider a specific proposal for a grammar of a fragment of English.

3.1 Trees

The derivation of complex expressions from lexical items is commonly represented with a type of graph called a *tree*. Part, *but not all*, of what we think of as the *structure* of an expression is given by its tree graph. For example, we might use the tree depicted in (1) to represent the English expression *John likes every teacher* (though the tree linguists would currently use is more complicated than (1) and would indicate that *likes* is present tense, as opposed to the simple past tense form *liked*).

(1)



This tree is understood to represent a variety of linguistic information. First, its bottommost items *John*, *likes*, *every*, and *teacher* are presented here as underived expressions (lexical items) having the categories indicated by the symbols immediately above them. Specifically, according to this tree (given here for illustrative purposes only), *John* is a DP (Determiner Phrase), *likes* is a TV (Transitive Verb), *every* is a Det (Determiner), and *teacher* is a N (Noun).

The tree in (1) identifies not only the lexical items *John likes every teacher* is constructed from, it also defines their **pronunciation order**. Specifically, we use the convention that the word written leftmost is pronounced first, then the next leftmost item, and so on. (Other writing conventions could have been used: in Hebrew and Arabic items written rightmost are pronounced first; in Classical Chinese reading may go from top down, not left to right or right to left).

Finally, (1) identifies which expressions combine with which others to form complex ones, resulting ultimately in the expression *John likes every teacher*. The expressions a derived expression is built from are its **constituents**. (1) also identifies the grammatical category of each expression. This in (1), *every* and *teacher* combine to form a constituent *every teacher* of category DP. The TV *knows* combines with this constituent to form another constituent, *knows every teacher* of category VP. And this in turn combines with *John* of category DP to form *John likes every teacher* of category S. Note that some substrings of the string *John likes every teacher* are not constituents of it according to (1): they were not used in building that particular S. For example, *knows every* is not a constituent, and that string has no category. Similarly, *John knows* and *John knows every* are not constituents of (1).

The tree in (1) does not exhibit the rules that applied to combine various words and phrases to form constituents. In the next chapter, we formulate some such rules. Here we just suggest some candidates so that the reader can appreciate the sense in which (1) records the derivational history of *John likes every teacher*, even though some structurally relevant information has been omitted.

To build the DP *every teacher* from the Det *every* and the N *teacher*, the simplest rule would be the concatenative one whose effect is given in (2):

(2) if s is a string of category Det and t is a string of category N, then $s \frown t$ is a string of category DP.

Recall that $s \frown t$ denotes the concatenation of the sequences s and t.

Similarly, we might derive the VP *knows every teacher* by concatenating *knows* of category TV with *every teacher* of category DP. And then we might derive the S *John likes every teacher* by concatenating *John* with that VP string.

Linguists commonly assume that the trees they use to represent the derivation of expressions are in fact derived by concatenative functions of the sort illustrated in (2). Such functions will take n expressions as arguments and derive an expression by concatenating the strings of those expressions, perhaps inserting some constant elements. For example, we might consider a function of two arguments, which would map *John* of category DP and *cat* of category N to *John's cat* of category DP. This function introduces the constant element *'s*.

We are not arguing here that the rules of a grammar—its structure building functions—*should* be concatenative, we are simply observing that linguists commonly use such functions. And this in turn has a limiting effect, often unintended, on how expressions can be syntactically analyzed and hence how they can be semantically interpreted. Here is an example which illustrates the use of a non-concatenative function. It introduces some non-trivial issues taken up in more detail in our later chapters on semantics.

Ss like (3) present a subtle ambiguity. (3) might be interpreted as in (a), or it might be interpreted as in (b).

(3) Some editor read every manuscript.
  a. Some editor has the property that he read every manuscript.
  b. Every manuscript has the property that some editor read it.

One the a-reading, a speaker of (3) asserts that there is at least one editor who read all the manuscripts. But the b-reading is weaker. It just says that for each manuscript, there is some editor who read it. Possibly different manuscripts were read by different editors. Thus in a situation in which there are just two editors, say Bob and Sue, and three manuscripts, say m_1, m_2, and m_3, and Bob read just m_1 and m_2, and Sue read just m_2 and m_3, we see that (3) is true on the b-reading: every manuscript was read by at least one editor. But (3) is false on the a-reading, since no one editor read all of the manuscripts. Ambiguities of this sort are known as *scope ambiguities* and are taken up in Chapter 8.

One approach to representing these ambiguities originates with the work of Montague (1974). This approach says that (3) is syntactically ambiguous – derived in two interestingly different ways. In one way, corresponding to the a-reading, it is derived by the use of concatenative functions as we have illustrated for (1). The difference this time is that the last step of the derivation concatenates a complex DP *some editor* with the VP *read every manuscript*; earlier we had used not a complex DP but rather the lexical DP *John*. The derivation of the S whose interpretation is the b-reading is more complicated. First we derive by concatenation a VP *read it* using the pronoun *it*. Then we concatenate that with the DP *some editor* to get the S *Some editor read it*. Then we form *every manuscript* by concatenation as before. In the last step, we derive *Some editor read every manuscript* by **substituting** *every manuscript* for *it*. So the last step in the derivation is a substitution step, not a concatenation step. It would take two arguments on the left in (4) and derive the string on the right.

(4) every ms, some editor read it $\Longrightarrow$ some editor read every ms

Let us emphasize that while the a-reading has a standard tree derivation, the b-reading does not, since *read every manuscript* is not formed solely by concatenative functions. Thus if we were to limit ourselves to the use of standard trees in representing derivations of natural language expressions, we would exclude some ways of compositionally interpreting semantically ambiguous expressions. For the record, let us formally define the core substitution operation.

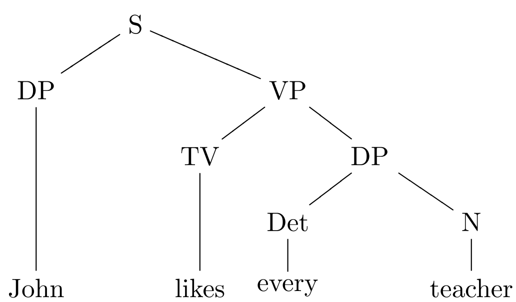
Definition 3.1. Let s be a string of length $n > 0$. Let t be any string, and let i be a number between 1 and n. Then $s(i/t)$ is the string of length n whose i^{th} coordinate is t and whose j^{th} coordinate is s_j, for all $j \neq i$. We call $s(i/t)$ the result of *substituting the* i^{th} *coordinate of* s *by* t.

Exercise 3.1. Complete the following in list notation.

a. $\langle 2, 5, 2\rangle(2/7) =$ ________.
b. $\langle 2, 2, 2\rangle(2/7) =$ ________.
c. $\langle \text{John}, \text{'s}, \text{cat}\rangle(3/\text{dog}) =$ ________.
d. $\langle \text{every}, \text{cat}\rangle(2/\text{fat cat}) =$ ________.

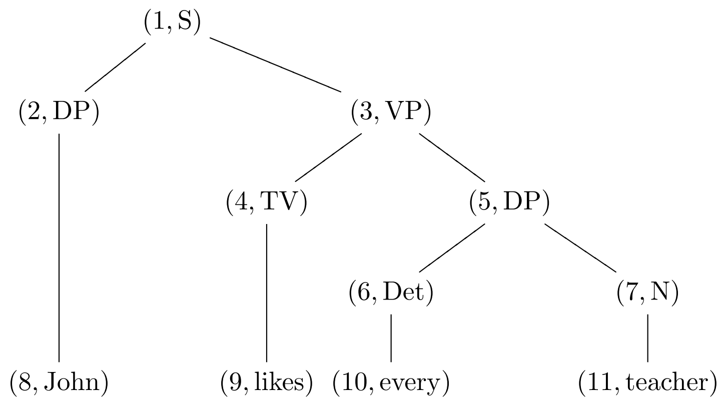
Trees as mathematical objects Having presented some motivation for the linguists' use of trees, we now formally define these objects and discuss several of the notions definable on trees that linguists avail themselves of. For convenience, we repeat the tree in (1).

(5)



The objects presented in (5) are linguistic *labels*—names of grammatical categories, such as S, DP, VP, etc., or English expressions such as *John*, *knows*, etc. These labels are connected by lines, called *branches* or *edges*. We think of the labels as labeling *nodes* (or *vertices*), even though the nodes are not explicitly represented. But note, for example, that the label DP occurs twice in (5), naming different nodes. The node representing the category of *John* is not the same as that representing the category of *every teacher*, even though these two expressions have the same category. So we must distinguish nodes from their labels, since different nodes may have the same label. In giving examples of trees below, we shall commonly use numbers as nodes, in which case (5) could receive a representation as in (6).

(6)



Now we can say that node 2 and node 5 have the same label, DP. Our formal definition of *tree* will include nodes, and their absence on specific occasions is just one more typical instance of simplifying a notation when no confusion results.

Now let us consider what is distinctive about the graph structure of trees. Some nodes as we see are connected to others by a line (a branch). If we can read *down* along branches from a node x to a node y, we say that x *dominates* y. And the distinctive properties of trees lie in the properties of this dominance relation. First, we understand that if we can move down from a node x to a node y (so x dominates y), and we can move down from y to a node z (so y dominates z), then we clearly can move down from x to z, whence x dominates z. Thus *dominates* is a *transitive* relation.

We have already seen one transitive relation, namely inclusion of subsets ($\subseteq$): given a collection of sets, we see that if $X \subseteq Y$ and $Y \subseteq Z$, then also $X \subseteq Z$. Many common mathematical relations are transitive. For example, the $\geq$ relation on natural numbers is transitive: if $n \geq m$ and $m \geq p$, then $n \geq p$. So let us define more generally.

Definition 3.2. R is a *binary relation* on a set A if R is a subset of $A \times A$. Instead of writing $(x, y) \in R$, we often write xRy, read as "x stands in the relation R to y."

In general, to define a binary relation R on a set A we must say for each choice x and each choice y of elements from A whether xRy or not. In particular this means that we must say for $x \in A$, whether xRx or not.

In what follows we are concerned with whether various relations we define of interest are *reflexive*, *antisymmetric*, *asymmetric*, or *transitive*. These notions are defined below.

Definition 3.3. A relation R on a set A is *transitive* if for all x, y, $z \in A$, if xRy and yRz, then xRz.

As we have seen, the *dominates* relation among nodes in a given tree is transitive. Further, *dominates* is "loop-free", meaning that we can never have two *different* nodes each of which dominates the other. The traditional term for "loop free" is *antisymmetric*:

Definition 3.4. A binary relation R on a set A is *antisymmetric* iff for $x, y \in A$, xRy and yRx jointly imply $x = y$. (Antisymmetry should not be confused with *asymmetry*, defined as follows: A binary relation R on a set A is *asymmetric* iff for x, $y \in A$, if xRy, then yRx is *false*. For example the proper subset relation $\subset$ is asymmetric: if A is a proper subset of B, B is certainly not a subset of A, hence not a proper subset of A. Similarly the 'is strictly less than' relation $<$ in arithmetic is asymmetric, as is the 'is a parent of' relation on people.

Again, $\subseteq$ is antisymmetric. If $X \subseteq Y$ and $Y \subseteq X$, then X and Y have the same members and are hence equal. Similarly, one checks that the arithmetical $\geq$ is antisymmetric.

Note that the antisymmetry of a relation R still allows that a given element x stand in the relation R to itself. In what follows, we treat *dominates* as a *reflexive* relation, meaning that each node is understood to dominate itself. For the record:

Definition 3.5. A binary relation R on a set A is *reflexive* iff for $x \in A$, xRx. R is *irreflexive* iff for $x \in A$, it is not the case that xRx. We write $\neg xRx$ in this case as well.

Now the cluster of properties that we have adduced for *dominates*, transitivity, antisymmetry, and reflexivity, is a cluster that arises often in mathematical study. We note:

Definition 3.6. A binary relation R on a set A is a *reflexive partial order* iff R is reflexive, transitive, and antisymmetric. The pair (A, R) is often called a *partially ordered set* or *poset*

In practice, when we refer to partial order relations, we shall assume that they are reflexive unless explicitly noted otherwise. Note that $\subseteq$ and $\geq$ are reflexive partial orders.

Exercise 3.2. Exhibit a binary relation R on $\{a, b, c\}$ which is neither reflexive nor irreflexive.

Exercise 3.3. We define a binary relation R on a set A to be *symmetric* if whenever xRy then also yRx. R is *asymmetric* if whenever xRy then it is not the case that yRx. Let R be a reflexive partial order on a set

A, and define another binary relation SR, *strict-R*, on A by

$$\text{for all } x,\, y \in A,\ xSRy \text{ iff } xRy \text{ and } x \neq y.$$

Prove that SR is irreflexive, asymmetric, and transitive. Such relations will be called *strict partial orders*.

For example, the strictly greater-than relation on numbers, $>$, is the strict-$\geq$ relation. So by this exercise, it is is a strict partial order. So is the strict-$\subseteq$ relation defined on any collection of sets. This relation is written $\subset$.

Exercise 3.4. Given a relation R on a set A, we define a relation R^{-1} on the same set A, called *the converse of* R by: $xR^{-1}y$ iff yRx, for all x, $y \in A$. Show that

a. If R is a reflexive partial order, so is R^{-1}.
b. If R is a strict partial order, so is R^{-1}.

In each case, state explicitly what is to be shown before you show it.

Returning to *dominates*, it has two properties that go beyond the partial order properties. First, it has a *root*, a node that dominates all nodes. In (6) it is node 1, the node labeled S.

Observation If R is a partial order relation on a set A, then there cannot be two distinct elements x and y such that each bears R to all the elements of A. The reason is that if both x and y have this property, then each bears R to the other. Hence by the antisymmetry of R, we have $x = y$. This shows that x and y are not distinct after all.

The second and more important property of the dominance order is that its branches never coalesce: if two nodes dominate a third, then one of those two dominates the other. We summarize these conditions below in a formal definition. The objects we define are unordered, unlabeled trees which we call *simple trees* (usually omitting 'simple'). We do not impose a left-right order on the bottommost nodes, and we do not require that nodes be labeled. For this reason, simple trees might well be called *mobiles*. Simple trees are ideal for studying pure constituent structure. Once they are understood, we add additional conditions to obtain the richer class of trees that linguists use in practice.

Definition 3.7. A *simple tree* T is a pair (N, D), where N is a set whose elements are called *nodes* and D is a binary relation on N called *dominates*, satisfying (a) - (c):

a. D is a **reflexive partial order** relation on N.
b. **the root condition**: There is a node r which dominates every node. This r is provably unique and called the *root* of T.

c. **chain condition** For all nodes x, y, z, if xDz and yDz, then either xDy or yDx. [1]

On this definition, the two pictures in (7) are the *same* (simple) tree. The difference in left-right order of node notation is no more significant then the left-right order in set names like '$\{2, 3, 5\}$'; it names the same set as '$\{5, 2, 3\}$'.

(7) 

Unordered trees are frequently used to represent structures of quite diverse sorts – chain of command hierarchies, classification schemes, genetic groupings of populations or languages. These often do not have a left-right order encoded. For example, Figure 3 is a tree representing the major genetic groupings of Germanic languages.

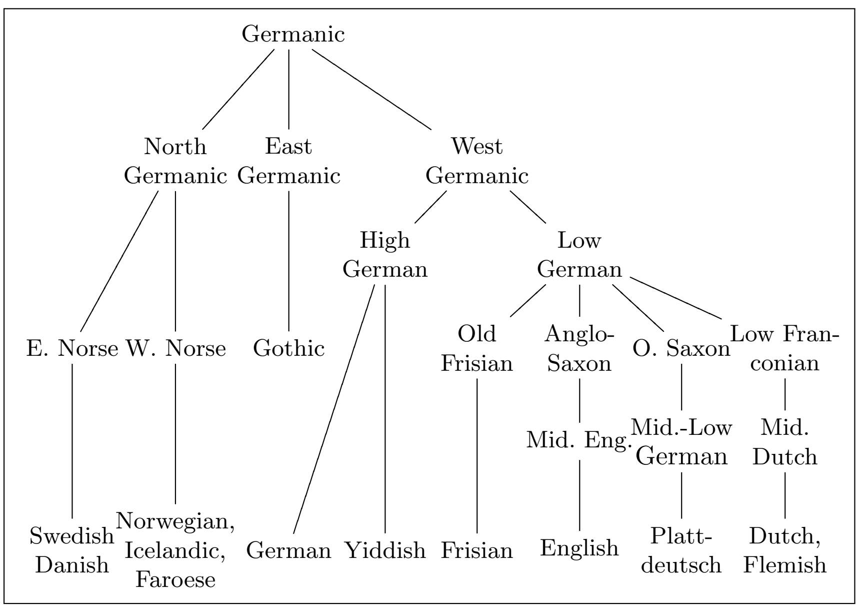


FIGURE 3 The major Germanic Languages

The root node of Figure 3 represents a language, Germanic, from which all other languages shown are genetically descended. The leaves

[1] The reason for this name is that the chain condition as defined above is equivalent to the assertion that the set of nodes dominating any given node is a chain, that is, linearly ordered, by D. For the definition of a linear order, see page 72.

(at the bottom) are languages or small groups of closely related languages. Notice that Gothic is also a leaf. To see how closely related two languages are, read up the tree until you find the first common ancestor. For example, (Modern) German and Yiddish are more closely related than either is to (Modern) English, since they are descended from High German, itself coordinate with Low German. So the least common ancestor of English and German is W. Germanic, itself a proper ancestor of High German.

And observe that the left-right order on the page of the names of the daughter languages has no structural significance. There is no sense in which Icelandic is to the left of, or precedes, English, or Dutch to the right of English

Let us consider in turn the three defining conditions on trees. The discussion will be facilitated by the following definitions:

Definition 3.8. Let $T = (N, D)$ be a tree. Then for all x, $y \in N$,

a. *x strictly dominates y*, $xSDy$, iff xDy and $x \neq y$.

b. *x immediately dominates y*, $xIDy$, iff x strictly dominates y, but there is no node z such that x strictly dominates z and z strictly dominates y.

In drawing pictures of trees, we just draw the *ID* relation. So in (6), $1ID2$ and $1ID3$, but $\neg 1ID4$. This last fact holds despite the fact that $1D4$ and indeed $1SD4$. Observe that when φ is a sentence, we sometimes write $\neg\varphi$ for the sentence "it is not the case that φ".

Definition 3.9. A tree $T = (N, D)$ is *finite* if its node set N is finite.

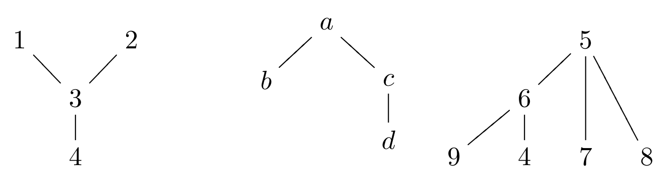
In this book we only consider finite trees.

Consider the dominance relation D on trees. Because D is an order relation (transitive and antisymmetric), we know that there can be no non-trivial *cycles*, that is, no two sequences of two or more nodes that begin and end with the same node and in which each node (except the last) immediately dominates the next one. Such a sequence couldn't have the form $\langle x, x, x, \ldots \rangle$ because no node x can immediately dominate itself (since then x would strictly dominate itself and hence be non-identical to itself). Nor could such a sequence have the form $\langle x, y, \ldots, x, \ldots \rangle$, with $x \neq y$, since then we could infer that xDy and yDx, whence $x = y$ by antisymmetry of D. This contradicts our assumption.

Second, linguists often don't consider the case where a given node might dominate itself. Usually when we speak of x dominating y, we are given that x and y are different nodes. In case where x and y are intended as different but not independently given as different, it would be clearer for the linguist to say "x strictly dominates y".

Third, our tree pictures do not include the transitivity edges – there is no edge directly from 1 to 4 in (6), for examples. Nor do we have to put in the reflexivity loops, the edges from each node to itself. We just represent the immediate dominance relation (sometimes called the *cover relation*), the rest being recoverable from this one by the assumptions of transitivity and reflexivity. Now, of the three conditions that the dominance relation D must satisfy, the root condition rules out relations like those with diagrams in (8):

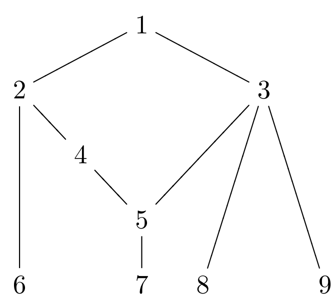
(8) (*a*) (*b*)



In (8a) there is clearly no root, that is no node that dominates every node. And (8b) is just a pair of trees. There is no root since no node dominates both a and 5. (A graph with all the properties of a tree except the root condition is sometimes called a *forest*.) So neither (8a) nor (8b) are graphs of trees.

The truly distinctive condition on trees, the one that differentiates them from many other partial orders, is the chain condition. Consider the graph in (9), as always reading down.

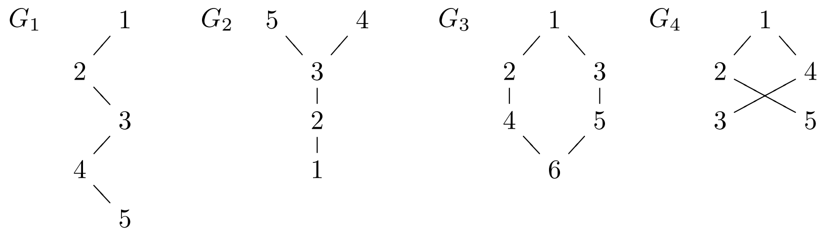
(9)



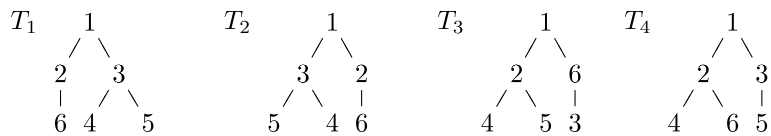
(9) violates the chain condition: for example, both 2 and 3 dominate 5, but neither 2 nor 3 dominates the other.

We present in Figure 4 a variety of linguistic notions defined on simple trees (and thus ones that do not depend on labeling or linear order of elements).

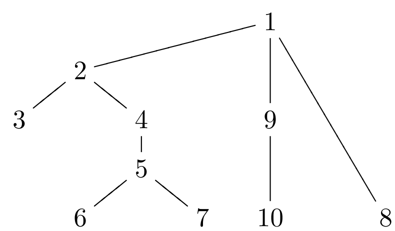
Exercise 3.5. For each graph below, state whether it is a tree graph or not (always reading down for dominance). If it is not, state at least one of the three defining conditions for trees which fails.



Exercise 3.6. Below are four graphs of trees, T_1, …, T_4. For each distinct i, j between 1 and 4, state whether $T_i = T_j$ or not. If not, give one reason why it fails.



Exercise 3.7. Referring to the tree below, mark each of the statements T (true) or F (false) correctly. If you mark F, say why.



a. 4 and 9 are sisters	b. $2SD7$	c. $1ID8$
d. 2 and 8 are sisters	e. 1 is mother of 8	f. 3 is a leaf
g. $depth(5) = 3$	h. $depth(T) = depth(7)$	i. $5IND7$
j. $depth(8) > depth(2)$	k. $depth(7) = depth(10)$	l. $2CC5$
m. $\langle 4, 2, 3, 2\rangle$ is a path	n. $\langle 7, 5, 4, 2, 3\rangle$ is a path	o. $5CC3$
p. $\langle 8\rangle$ is a path		

q. 8 asymmetrically c-commands 3.

r. For all nodes x, y, if x is mother of y, then y is mother of x.

We conclude with an important fact about trees:

Theorem 3.1. *If x and y are distinct nodes in a tree T, then there is exactly one path from x to y in T.*

Let $T = (N, D)$ be a tree, and let x and y be nodes of N.

a. x is an *ancestor* of y, or, dually, y is a *descendant* of x, iff xDy.
b. x is a *leaf* (also called a *terminal node*) iff $\{z \in N \mid xSDz\} = \emptyset$.
c. the *degree* of x, noted $deg(x)$, is the size of $\{z \in N \mid xIDz\}$. (Some texts write *out-degree* where we write simply degree). So if z is a leaf then $deg(z) = 0$.
d. x is a *n-ary branching node* iff $|\{y \in N \mid xIDy\}| = n$. We write *unary* branching for 1-ary branching and *binary* branching for 2-ary branching. T itself is called *n-ary branching* if all nodes except the leaves are n-ary branching. In linguistic parlance, a *branching* node is one that is n-ary branching for some $n \geq 2$. (So unary branching nodes are not called branching nodes by linguists).
e. x is a *sister* (*sibling*) of y iff $x \neq y$ and there is a node z such that $zIDx$ and $zIDy$.
f. x is a *mother* (*parent*) of y iff $xIDy$; Under the same conditions we say that y is a *daughter* (*child*) of x.
g. The depth of x, noted $depth(x)$, is $|\{z \in N \mid zSDx\}|$.
h. $Depth(T) = \max\{depth(x) \mid x \in N\}$. This is also called the *height* of T. Note that $\{depth(x) \mid x \in N\}$ is a finite non-empty subset of $\mathbb{N}$. (Any finite non-empty subset K of $\mathbb{N}$ has a greatest element, noted $\max(K)$.)
i. x is (dominance) *independent of* y iff neither dominates the other. We write *IND* for *is independent of*. Clearly *IND* is a symmetric relation. This relation is also called *incomparability*.
j. A *branch* is a pair (x, y) such that $xIDy$.
k. p is a *path* in T iff p is a sequence of two or more distinct nodes such that for for all i, $1 \leq i < |p|$, $p_i IDp_{i+1}$ or $p_{i+1}IDp_i$.
l. x *c-commands* y, noted $xCCy$, iff
   i. x and y are independent, and
   ii. every branching node which strictly dominates x also dominates y.

   We say that x *asymmetrically c-commands* y iff x c-commands y but y does not c-command x.

FIGURE 4 Linguistic notions defined on trees.

The topic of trees is standard in mathematics and computer science books (these will be on topics like graph theory, discrete mathematics or data structures). But there, the basic definition is often graph theoretic: one takes vertices and symmetric edges as primitive, and the definition of a tree is as in Theorem 3.1: between every two vertices there is a unique path. One can also go the other way, starting with a tree in the graph-theoretic sense, specify a root, and then recover the dominance relation.

3.2 C-command

Figure 4 defines a number of concepts pertaining to trees. Perhaps the only one of these that originates in linguistics is c-command. We want to spell out in detail the motivations for this concept. Here is one: Reflexive pronouns (*himself*, *herself*, and a few other *self* forms) in Ss like (10) are referentially dependent on another DP, called their antecedent.

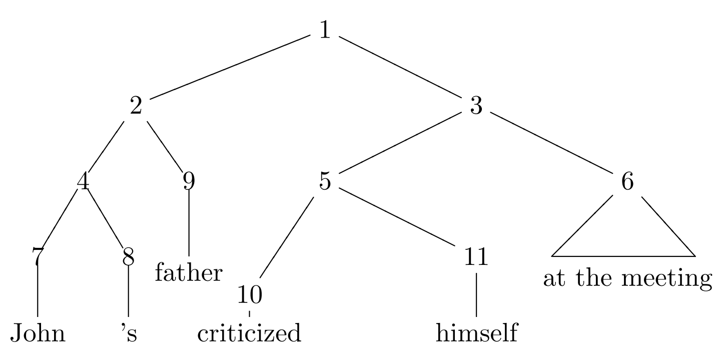
(10) *John's father embarrassed himself at the meeting.*

In (10) *John's father* but not *John* is the antecedent of *himself*. That is, (10) only asserts John's father was embarrassed, not John. *A linguistic query*: Given a reflexive pronoun in an expression E, which DPs in E can be interpreted as its antecedent? (11) is a *necessary* condition for many expressions:

(11) Antecedents of reflexive pronouns c-command them.

Establishing the truth of a claim like (11) involves many empirical claims concerning constituent structure which we do not undertake here. Still, most linguists would accept (12) as a gross constituent analysis of (10). (We "cover" the proper constituents of *at the meeting* with the widely used "triangle", as that internal structure is irrelevant to the point at hand).

(12)



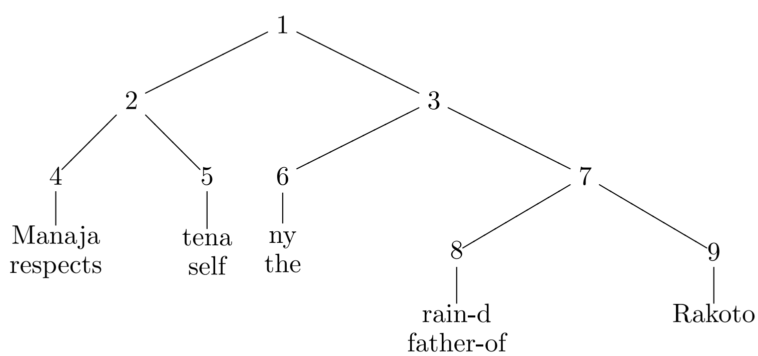
We see here that node 2, *John's father*, does c-command node 11, *himself*. Clearly 2 and 11 are independent, and every branching node which strictly dominates 2 also dominates 12 since the only such node is the root 1. In contrast, node 7, *John* does not c-command 11, since both 2 and 4 are branching nodes which strictly dominate 7 but do not dominate 11.

One might object to (11) as a (partial) characterization of the conditions regulating the distribution of reflexives and their antecedents on the grounds that there is a less complicated (and more traditional) statement that is empirically equivalent but only uses left-right order:

(13) Antecedents of reflexive pronouns precede them

In fact for basic expressions in English the predictions made by (11) and (13) largely coincide since the c-commanding DP precedes the reflexive[2]. But in languages like Tzotzil (Mayan: see Aissen (1987)) and Malagasy (Malayo-Polynesian; Keenan (1995)) in which the basic word order in simple active Ss is VOS (Verb + Object + Subject) rather than SVO (Subject + Verb + Object) as in English, we find that antecedents follow reflexives but still c-command them. So analogous to (10), speakers of Malagasy understand that (14) only asserts that Rakoto's father respects himself, but says nothing about Rakoto himself. So c-command wins out in some contexts in which it conflicts with left-right order.

(14)



[2]Some known empirical problems with the c-command condition are given by:

i. *It is only himself that John admires.*
ii. *Which pictures of himself does John like best?*
iii. *The pictures of himself that John saw in the post office.*

But these expressions are derivationally complex. It may be that c-command holds in simple expressions (e.g., *John admires only himself*) and that the antecedent-reflexive relation is preserved under the derivation of more complex ones.

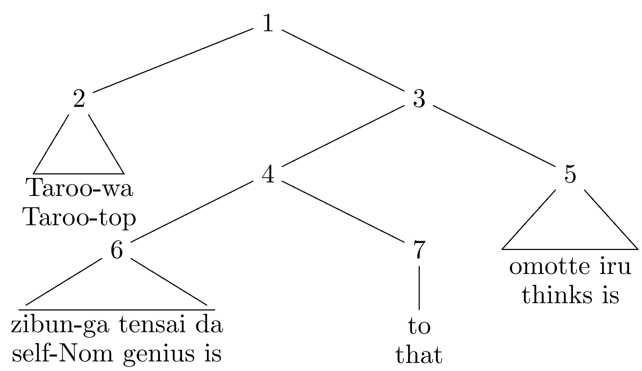
Rakoto's father respects himself (Malagasy)

Pursuing these observations it is natural to wonder whether c-command is a *sufficient* condition on the antecedent-reflexive relation. That is, can any DP which c-commands a reflexive be interpreted as its antecedent? Here the answer is a clear negative, though moreso for Modern English than certain other languages. Observe first that in (15a), the DP *every student* is naturally represented as a sister to the VP *thinks that Mary criticized himself* and

(15) a. **Every student thinks that Mary criticized himself*
b. **Every student thinks that himself criticized John*

And since *himself* lies properly within that VP, we have that *every student* (asymmetrically) c-commands *himself*. But it cannot be interpreted as its antecedent. Comparable claims hold for (15b). But patterns like those in (15), especially (15b), are possible in a variety of languages: Japanese, Korean, Yoruba, even Middle English and Early Modern English:

(16) (Japanese)



Taroo thinks that he (Taroo) is a genius.

(17) . . . a Pardonere . . . seide that hymself myghte assoilen hem alle Piers Plowman c.1375 . . . a Pardoner . . . said that himself might absolve them all Keenan (2007)

(18) he . . . protested . . ., that himselfe was cleere and innocent *Dobson's Drie Bobbes*, 1607. Keenan (2007)

(19) But there was a certain man, . . . which . . . bewitched the people of Samaria, giving out that himself was some great one (*King James Bible*, Acts 8.9, 1611)

So the possible antecedents for a reflexive pronoun in English thus appear to be a subset of the c-commanding DPs with the precise de-limitation subject to some language variation. See Büring (2005) for an overview discussion.

Exercise 3.8. For each condition below exhibit a tree which instantiates that condition:

a. CC is not symmetric.
b. CC is not antisymmetric.
c. CC is not transitive.
d. CC is not asymmetric.

In each case say why the trees show that CC fails to have the property indicated. We note regarding part (c) that asymmetric c-command is a transitive relation.

Exercise 3.9. In any tree,

a. if $aCCb$ and bDx does $aCCx$?
b. Do distinct sisters c-command each other?
c. c-command is irreflexive. Why?
d. For all nodes a, $\{x \in T \mid xDa\} \neq \emptyset$. Why?

3.3 Sameness of Structure: Isomorphism

Our interest in trees concerns the structural relations between nodes – relations defined in terms of dominance – not the identity of the nodes themselves. For example the tree T_1 below whose nodes are the numbers 1 through 5 and T_2 whose nodes are the letters a through e are regarded as "essentially" the same. They have the same "branching structure", differing just by identity of nodes. And these, as we have noted, are normally not even noted in tree graphs used by linguists.

Thus we want away of saying that T_1 and T_2 have the same structure, are *isomorphic*, even though they fail to be identical. Then any structural claim we can make of one will hold of the other as well. For example the statement "All non-terminal nodes are binary branching" holds of both; "The total number of nodes is 9" fails of both trees. But no structural statement can hold of one but fail of the other.

Here is the core idea of isomorphism (an idea that generalizes naturally to other types of structures such as boolean algebras, groups, etc. and so is not peculiar to trees): Trees T and T' are isomorphic iff

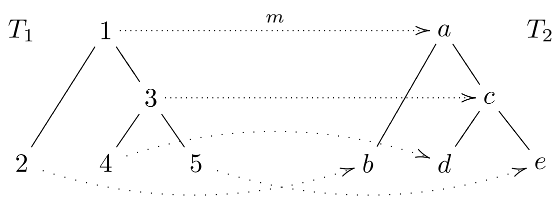
(1) we can match up their nodes one for one with none left over and (2) whenever a node x dominates a node y in one tree the node x is matched with dominates the one y is matched with in the other tree, and conversely. Formally:

Definition 3.10. A tree $T = (N, D)$ is *isomorphic* ($\cong$) to a tree $T' = (N', D')$ iff there is a bijective function m from N to N' satisfying:

$$\text{for all } x, y \in N,\ xDy \text{ iff } m(x)D'm(y).$$

Such a bijection is called an *isomorphism* (from T to T').

Thus, to prove that T_1 above is isomorphic to T_2, we must show that there is a bijection from the nodes of T_1 to those of T_2 satisfying the condition in Def 3.10. We do this by exhibiting such a bijection, m below as a dotted arrow.

(20)


To establish that the m shown in (20) is an isomorphism, we must verify (1) that m is a bijection, and (2) that m satisfies the condition in Def. 3.10. Visual inspection establishes that m is a bijection. To visually establish that m strongly preserves dominance check first that whenever xDy in T_1 than $m(x)$ dominates $m(y)$ in T_2 . Then you must check the converse: whenever x' dominates y' in T_2 then $m^{-1}(x')$, the node in T_1 that m maps to x', dominates $m^{-1}(y')$ in T_1 . This verifies that T_1 and T_2 have the same dominance structure.

Exercise 3.10. Let $\mathbf{T}$ be any collection of trees. Each statement below is true. Say why.

a. For all $T \in \mathbf{T}$, $T \cong T$.
b. For all $T, T' \in \mathbf{T}$, if $T \cong T'$ then $T' \cong T$.
c. For all $T, T', T'' \in \mathbf{T}$, $T \cong T'$ and $T' \cong T''$, then $T \cong T''$.

When two relational structures are isomorphic they have the same structurally definable properties. In particular, if two trees are isomorphic then they have the same tree definable properties. For example,

Fact 1 Let $T = (N, D)$ and $T' = (N', D')$ be isomorphic trees, let h be an isomorphism from T to T'. Then, for all $a, b \in N$:

a. $aSDb$ iff $h(a)SD'h(b)$.
b. $aIDb$ iff $h(a)ID'h(b)$.

Basic Facts About Isomorphisms

Let (A, R) and (B, S) be relational structures (So R is a binary relation defined on the set A, and S is a binary relation defined on B). Then (A, R) is isomorphic to (B, S), noted $(A, R) \cong (B, S)$, iff there is a bijection h from A into B satisfying

$$\text{for all } x, y \in A,\ xRy \text{ iff } h(x)Sh(y).$$

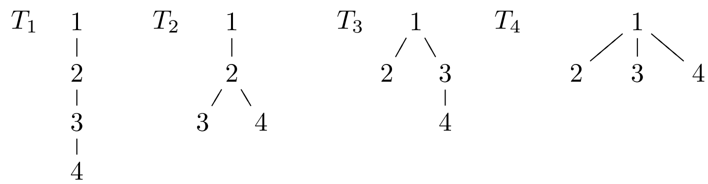
Such an h is called an *isomorphism* (from (A, R) to (B, S)).

a. If h is an isomorphism from (A, R) to (B, S) then h^{-1} is an isomorphism from (B, S) to (A, R).
b. If h is an isomorphism from (A, R) to (B, S) and g is an isomorphism from (B, S) to some (C, T) then $g \circ h$ is an isomorphism from (A, R) to (C, T).
c. Every relational structure (A, R) is isomorphic to itself, using the identity map $id_A : A \to A$. This map is defined by $id_A(a) = a$ for $a \in A$.

c. $deg(a) = deg(h(a))$.
d. a is 3-ary branching iff $h(a)$ is 3-ary branching.
e. $leaf(a)$ iff $leaf(ha)$.
f. $depth(a) = depth(ha)$
g. $aINDb$ iff $h(a)INDh(b)$.
h. $h(root(T)) = root(T')$.
i. $aCCb$ iff $h(a)CCh(b)$.
j. a and b are sisters iff $h(a)$ and $h(b)$ are sisters.
k. $|N| = |N'|$.

Remark You don't really know what the structures of a given class are until you can tell when two such are isomorphic. Using the fundamental fact that isomorphic structures make the same sentences true we see that trees T_1 and T_2 below are not isomorphic. T_2 for example has one node of out-degree 2, T_1 has no such node.

Fact 2 If T is a simple tree with exactly four nodes, then T is isomorphic to exactly one of the following:



Exercise 3.11.

a. In (20) we exhibited an isomorphism from T_1 to T_2 . Exhibit another isomorphism from T_1 to T_2 and conclude that there may be more than one isomorphism from one structure to another (in fact a very common case).

b. Exhibit a set of five-node trees with the following two properties:
   i. no two of them are isomorphic, and
   ii. any tree with exactly five nodes is isomorphic to one you have exhibited (Hint: the set you want has exactly 9 members).

3.3.1 Constituents

We turn to the important definition of *constituent*.

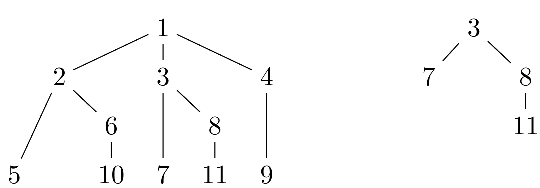
Definition 3.11. Let $T = (N, D)$ be a tree. For each node b of T, we define $T_b =_{df} (N_b, D_b)$, where

i. $N_b =_{df} \{x \in N \mid bDx\}$,
ii. for all $x, y \in N_b$, xD_by iff xDy.

We show that each T_b as defined is a tree, called the *constituent of* T *generated by* b. (Note already that N_b is never empty. Why?)

For example, consider the tree T depicted on the left in (21); T_3 is depicted on the right in (21):

(21)



Exercise 3.12. Using the T exhibited on the left in (21), exhibit

a. T_2
b. T_{10}
c. T_1

Theorem 3.2. *Let $T = (N, D)$ be a tree. For all $b \in N$, $T_b = (N_b, D_b)$ is a tree whose root is b.*

Definition 3.12. For all trees $T = (N, D)$ and $T' = (N', D')$, T' *is a constituent of* T ($T'\,CON\ T$) iff for some $b \in N$, $T' = T_b$.

Theorem 3.3. *Consider the set $T(\mathbb{N})$ of finite trees (N, D) with $N \subseteq \mathbb{N}$. CON, the "is a constituent of" relation defined on $T(\mathbb{N})$ is a reflexive partial order relation.*

Remark Our mathematically clear and simple definition of *constituent* should not be confused with the empirical issue of identifying the constituents of any given expression in English. This is often far from obvious. Here are a few helpful rules of thumb given just so the reader can see that our examples of constituents are not utterly arbitrary. Suppose that an expression s is a constituent of an expression t. Then, (1) s is usually semantically interpreted (has a meaning). (2) s can often be replaced by a single lexical item. (3) s has a grammatical category and can often be replaced by another expression of the same category. And (4), s will usually form boolean compounds in *and*, *or*, and *neither* ... *nor* ... with other expressions of the same category.

3.4 Labeled Trees

We now enrich the tree structures we have been considering to include ones whose nodes are labeled. The basic idea of the extension is fairly trivial; it becomes more interesting when the set of labels itself has some structure (as it does in all theories of grammar).

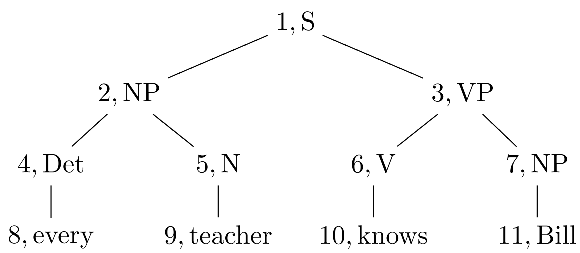
Definition 3.13. T is a *labeled tree* iff T is an ordered triple (N, D, L) satisfying:

i. (N, D) is a simple tree, and
ii. L is a function with domain N.

Terminology For $x \in N$, $L(x)$ is called the *label* of x. When we say that a labeled (unordered) tree is a triple we imply that to define such an object there are three things to define: a set N of nodes, a dominance relation D on N, and a function L with domain N.

Graphically we represent an (unordered) labeled tree as we represented unlabeled ones, except now we note next to each node b its label, $L(b)$:

(22)



So the labeled tree represented in (22) is that triple (N, D, L), where $N = \{1, 2, \ldots, 11\}$, D is that dominance relation on N whose immediate dominance relation is graphed in (22), and L is that function with domain N which maps 1 to 'S', 2 to 'DP', ..., and 11 to 'Bill'.

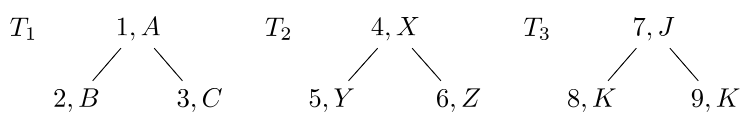
Labeled bracketing One often represents trees on the page by labeled bracketing, flattening the structure, forgetting the names of the nodes of the tree, and showing only the labels. For example, the labeled bracketing corresponding to (22) is

$$[[[\text{every}]_{\text{Det}}[\text{teacher}]_{\text{N}}]_{\text{NP}}[[\text{knows}]_{\text{V}}[\text{Bill}]_{\text{NP}}]_{\text{VP}}]_{\text{S}}.$$

Given our discussion above, a natural question here is "Under what conditions will we say that two (unordered) labeled trees are isomorphic?" And here is a natural answer, one that embodies one possibly non-obvious condition:

(23) h is an *isomorphism* from $T = (N, D, L)$ to $T' = (N', D', L')$ iff
  a. h is an isomorphism from (N, D) to (N', D') and
  b. for all $a, b \in N$, $L(a) = L(b)$ iff $L'(h(a)) = L'(h(b))$.

Condition (23a) is an obvious requirement; (23b) says that h maps nodes with identical labels to ones with identical labels and conversely. It guarantees for example that while T_1 and T_2 below may be isomorphic, neither can be isomorphic to T_3:



The three trees obviously have the same branching structure, but they differ in their labeling structure. In T_3, the two leaf nodes have the same label, 'K', whereas the two leaf nodes of T_1 (and also of T_2) have distinct labels. Hence no map h which preserves the branching structure can satisfy condition (23b) above, since h must map leaf nodes to leaf nodes and hence must map nodes with distinct labels to ones with the same label.

A deficiency with (23), however, is that all current theories of generative grammar use theories in which the set of category labels is highly structured. But we have not committed ourselves to any particular linguistic theory, only considering the most general case in which nodes are labeled, but no particular structure on the set of labels is given. When such structure is given, say the set of labels itself is built by applying some functions to a primitive set of labels, then that structure too must be fixed by the isomorphisms.

Below we consider informally one sort of case based on work in GB (Government & Binding) theory Lasnik and Uriagereka (1988). Within GB theory category labels (we usually just say "categories") are partitioned into *functional* ones and *content* ones. The latter include Ns (like *book* and *mother*), Vs (like *sleep* and *describe*), Ps (like *for* and *to*) and As (like *bold* and *bald*). The former include categories of "grammatical" morphemes like *Poss* for the possessive marker *'s* (as in *John's book*) or I for the inflection which marks tense and person/number on verbs, such as the *is* in *John is running*, or the *will* in *John will sleep.*

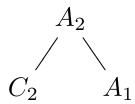
Cross classifying with the functional content distinction is a "bar level" distinction. A basic category C comes in three bar levels: C_0, C_1, and C_2. The bar level of a category pertains to the internal complexity of an expression having that category. Thus C_0's, categories of bar level zero, are the simplest. Expressions of zero level categories are usually single lexical items like *book* and *sleep*, or grammatical morphemes like *'s* and *will.* C_2's, categories of bar level 2, are complete phrasal expressions. For example *John will sleep* and *John's cat* have (different) categories of bar level 2. A category X of bar level 2 is called a *phrasal* category and noted XP.

Phrasal categories combine with categories of bar level 0 to form ones of bar level one according to the tree schema below (nodes suppressed, as is common practice).

(24)
$$\begin{array}{ccc} & A_1 & \\ / & & \backslash \\ A_0 & & B_2 \end{array}$$

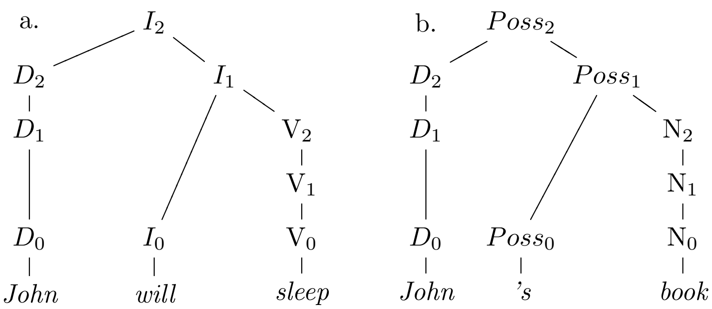
The expression of category A_0 in (24) is called the *head* of the entire A_1, and the expression of category B_2 is called its *Complement.* An example is the V_1 *describe the thief* whose head is the V_0 *describe* and whose complement is *the thief.* Similarly, *in the garden* is a P_1 headed by the P_0 *in.* A second type of labeled tree accepted by GB grammars is illustrated in (25).

(25)



A category of level 2 which is sister to a one level category as in (25) is called the *Specifier* of the entire expression. The head of an expression like (25) is the head of the A_1 expression. In the next figure, we exhibit two expressions each illustrating (24) and (25), noting that we allow that Specifiers and Complements may be absent.

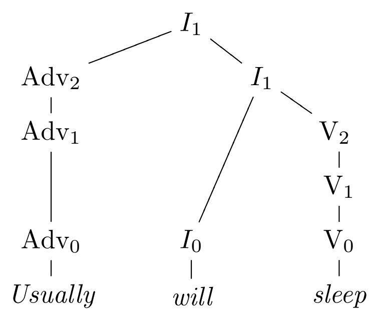
(26)



These two expressions have different categories. (26a) is an I_2, that is, an Inflection Phrase (IP) and (26b) is a $Poss_2$, that is, a Possessive Phrase ($PossP$). It is easy to see that (26a) and (26b) are isomorphic. Having drawn the graphs to scale we can superpose (26a) and (26b) in such a way that (1) the branching structures coincide and (2) the bar levels of labels on matching nodes coincide and (3) the labels on nodes in (26a) are distinct iff the ones they are matched with in (26b) are distinct. Note that this last condition does not follow from the others. Suppose for example that we replaced the label N_0 in (26b) with D_0 (a replacement which is not in fact sanctioned by GB grammars). Then we have not changed branching structure nor bar level of labels (since we replaced a zero level label with a zero level one) but now the distinct D_0 and V_0 nodes in (26a) correspond to two distinct D_0 nodes in (26b); that is, nodes with distinct labels are matched with ones having the same label and thus the trees are not isomorphic.

In addition to trees whose labels satisfy the schema in (24) and (25), we also find GB trees like those in (27), called *Adjunction structures* (Adv_2 is read "Adverb Phrase"):

(27)



(27) has the same branching structure as (26a) and (26b). So if the labeling on these trees were erased the resulting unlabeled trees would be isomorphic. But none of those isomorphisms can preserve distinctness of node labels or their bar level. Any isomorphism from (26a) to (27) must map the root to the root and hence associate a 2 level label with a 1 level one. And since it must map daughters of the root to daughters of the root, it cannot preserve label distinctness since both root daughters in (26a) have different labels from the root label. But this is not so in (27).

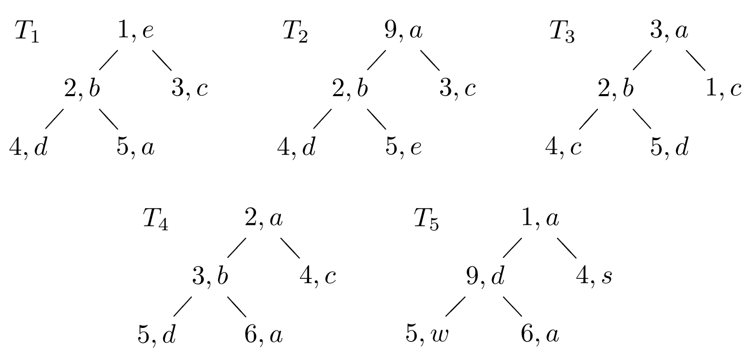
We see, then, that if h is an isomorphism from a GB tree $T = (N, D, L)$ to a GB tree $T' = (N', D', L')$, then, in addition to the conditions in (23), we should require:

(28) For all nodes x of T,

a. the bar level of $L(x)$ = the bar level of $L'(hx)$, and

b. $L(x)$ is a functional category iff $L'(hx)$ is a functional category.

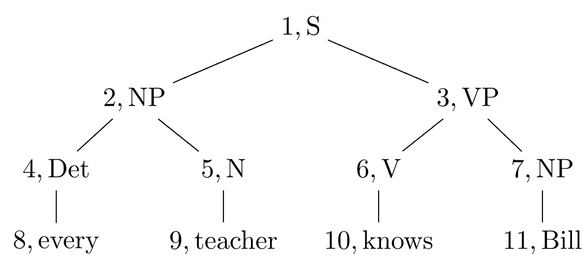
Exercise 3.13. For all distinct T, T' in the set of (unordered) labeled trees below, exhibit an isomorphism between them if they are isomorphic, and give at least one reason why they are not isomorphic if they are not. (The nodes are exhibited to facilitate your task).



3.5 Ordered Trees

As already noted, linguists use labeled trees to represent the pronunciation order of expressions. Pronounced expressions are represented by the labels on the leaf nodes of trees, and the pronunciation order is given by the left-right order in which the labels on leaf nodes are written on the page: the leftmost expression is pronounced first, then the second leftmost, etc. Thus in ordinary usage the tree graph in (22), repeated below as (29), not only represents constituents and their labels, it also tells us that *every* is pronounced before *knows*, *knows* before *Bill*, etc.

(29)



We consider more precisely the properties of the pronunciation order of expressions. Clearly it is transitive: if x is pronounced before y, and y before z, then, obviously, x is pronounced before z. It is also clearly asymmetric: if x is pronounced before y, then y is not pronounced before x. (This also means that antisymmetry holds, albeit vacuously).

Let us write simply '$<$' for the left-right order on the leaf nodes of a tree. When $x < y$ we say that x *precedes* y or that y *follows* x. And observe that for any two distinct leaves one must precede the other. That is, $<$ is a *total* (synonym: *linear*) order of the leaf nodes. Here is the definition:

Definition 3.14. A binary relation R on a set A is a *linear (total) order* iff

i. R is transitive, and
ii. R is antisymmetric, and
iii. R is total (that is, for all $x \neq y \in A$, xRy or yRx).

Examples Clearly $\leq$ in arithmetic is a linear order. We have already seen that it is transitive and antisymmetric. And for totality we observe that for any distinct numbers m and n, either $m < n$ or $n < m$. Also the strictly less than relation, $<$, is a linear order. It is obviously transitive. Since it is asymmetric ($n < m$ implies $\neg(m < n)$), it is antisymmetric.

And it is total: for distinct m and n, either $m < n$ or $n < m$. In contrast the subset relation $\subseteq$ defined on $\mathcal{P}(A)$ (the power set of A) for A with at least two distinct elements, say a and b, ..., is not total. There are subsets of A such that neither is a subset of the other. For example $\{a\}$ and $\{b\}$ have this property. So in general the subset relation on a collection of sets is a properly partial order, not a total (or linear) order. Note further, in analogy to $>$, that the proper subset relation is irreflexive, asymmetric and transitive, and again, normally not a total order.

Using these notions we define the notion *ordered tree*. We take a conservative approach at first, defining a larger class of ordered trees than is commonly considered in linguistic work. Then we consider an additional condition usually observed in the linguistic literature but which rules out some trees which seem to have some utility in modeling properties of natural language expressions.

Definition 3.15. a. $T = (N, D, L, <)$ is a *leaf ordered labeled tree* (or lol tree) iff

i. (N, D, L) is a labeled tree, and
ii. $<$ is a strict linear order of the terminal nodes.

The graphical conventions for representing lol trees are those we have been using, with the additional proviso that the left-right written order of leaf labels represents the precedes order $<$. The notions we have defined on trees in terms of dominance carry over without change when passing from mere unordered or unlabeled trees to lol trees. Only the definition of "constituent" needs enriching in the obvious way. Each node b of a tree T determines a subtree T_b, the constituent generated by b, as before, only now we must say that nodes of the subtree have the same labels they have in T and the leaves of the subtree are linearly ordered just as they are in T. Formally,

Definition 3.16. Let $T = (N, D, L, <)$ be a lol tree. Then for all $b \in N$, $T_b =_{df} (N_b, D_b, L_b, <_b,)$, where

$$\begin{array}{ll} N_b = \{x \in N \mid bDx\} & L_b(x) = L(x), \text{ all } x \in N_b \\ D_b = D \cap (N_b \times N_b) & <_b = \{(x, y) \mid x, y \in N_b \text{ and } x < y\} \end{array}$$

And one proves that T_b is a lol tree, called, as before, the *constituent generated by b*.

An additional useful notion defined on lol trees is that of the *leaf sequence* of a node. This is just the sequence of leaves that the node dominates. It is often used to represent the constituent determined by the node. Formally we define:

Definition 3.17. For b a node in a lol tree T, $LS(b)$ or the *leaf sequence determined by* b, is the sequence $\langle b_1, \ldots, b_n \rangle$ of leaves which b dominates, listed in the $<$ order.

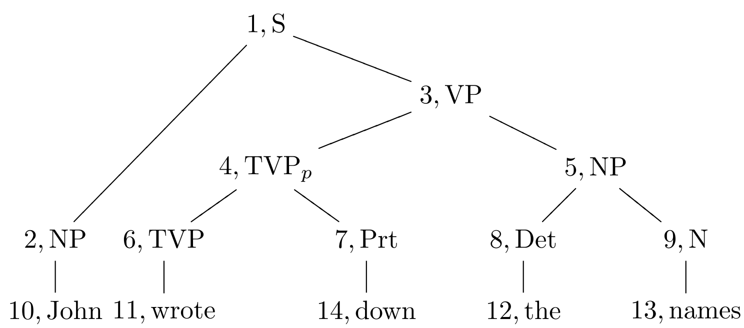
That is, the leaf sequence of a node is the string of leaf nodes it dominates.

What we have defined as lol trees differ from the type of tree most widely used in generative grammar in that we limit the precedes order $<$ to leaves. It is the labels of leaves which represent the words and morphemes that are actually pronounced and our empirical judgments of pronunciation order are highly reliable. Now the $<$ order on the leaf nodes extends in a straightforward way to certain internal (non-leaf) nodes as follows:

Definition 3.18. For x, y nodes of a lol tree T, $x <^* y$ iff every leaf node which x dominates precedes ($<$) every leaf node that y dominates.

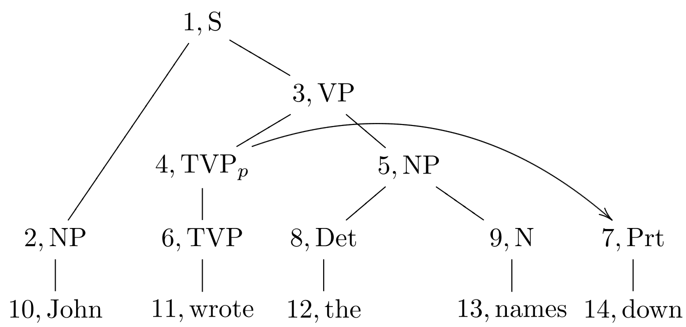
Note that when x and y are leaves, then, $x <^* y$ iff $x < y$ since the leaf nodes that x dominates are just x and those that y dominates are just y. When being careful, we read $<^*$ as *derivatively precedes*. But most usually we just say *precedes*, the same as for the relation $<$. By way of illustration consider (30), reading ‘Prt’ as particle:

(30)



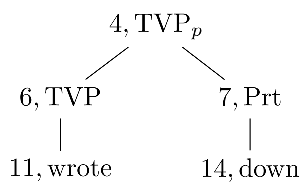
Here 4 precedes ($<^*$) 12, since every leaf that 4 dominates, namely 11 and 14, precedes every leaf that 12 dominates (just 12 itself). Equally 4 precedes 5, 8, 9, and 13. But 4 does not precede 7: it is not so that every leaf 4 dominates precedes every leaf that 7 dominates since 4 dominates 14 and 7 dominates 14, but 14 does not precede 14 since $<$ is irreflexive. Observe now that (31) is also an lol tree:

(31)

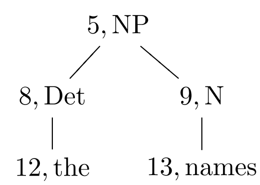


The lol trees in (30) and (31) are different, though they have the same nodes and each node has the same label in each tree. They also have identical dominance relations: n dominates m in (30) iff n dominates m in (31). But they have different precedence relations since in (30), 14 precedes 5 and everything that 5 dominates, such as 12 and 13. But in (31), 14 does not precede 5 or anything that 5 dominates. In consequence the constituents of (30) are not exactly the same as those of (31), though there is much overlap. For example (32a) is a constituent of both (30) and of (31). So is (32b)

(32) a.



b.



Exercise 3.14. Exhibit the smallest constituent of (30) that is not a constituent of (31).

The constituent T_4 in (31) is a classical example of a *discontinuous constituent*: its sequence of leaf nodes $\langle 11, 14\rangle$ is not a subsequence of the leaf sequence $\langle 10, 11, 12, 13, 14\rangle$ of the entire tree. (Recall that a sequence s is a subsequence of a sequence t iff there are sequences u,v (possibly empty) such that $t = usv$). Formally,

Definition 3.19. For all lol trees T and T', T' is a *discontinuous constituent* of T iff T' is a constituent of T and the leaf sequence of T' is not a subsequence of the leaf sequence of T.

Note that we have here defined a binary relation between trees: *is a discontinuous constituent of*. Whether a tree like (32a) is a discontinuous constituent of a tree T depends crucially on the relative linear order of leaves of (32a) with the leaves of T. We cannot tell just by looking at a tree T' in isolation whether it is discontinuous or not.

Most work in generative grammar does not countenance discontinuous constituents so our understanding of the role they might play in linguistic description, and theory, is limited. Still, drawing on Blevins (1994), Huck and Ojeda (1987), McCawley (1988, 1982), Ojeda (1987), and Blevins (1994) here are three phenomena whose representation by discontinuous constituents is prima facie plausible.

i. **cooccurrence restrictions.** The simplest cases here are ones in which the possibility of occurrence of a certain word depends on the presence of another. This is naturally accounted for if the two words are introduced as a single unit, though perhaps presented in non-adjacent positions. In fact the Verb+Particle construction in (30) and (31) illustrates this case. The choice of particle, *down* in our examples, depends significantly on the choice of verb: we do not say **John printed/erased/memorized/forgot the names down.* One way to represent this would be to treat the ordered pair $\langle(write, \mathrm{TV}), (down, \mathrm{PRT})\rangle$ as a complex lexical item, the rules which combine it with a DP object like 'the names" being defined in such a way as to allow the particle on either side of the DP . Formalism aside, the effect of the rules would be:

(33) If $\langle x, y\rangle$ is a TVP-Particle pair and z is a DP, then
  i. xzy is a string of category VP, and
  ii. if z is not a pronoun, xyz is a string of category VP.

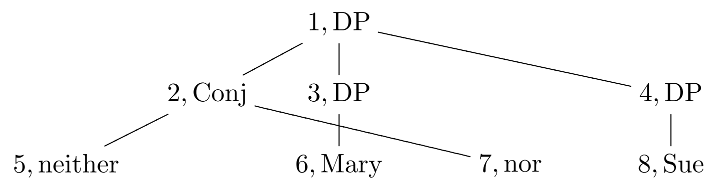
(The condition ii. blocks generating strings like **write down them*).

There are many other sorts of lexical cooccurrence restrictions in English. For example observe that in humble coordinations we find that the presence of *both*, *either*, and *neither* conditions the choice or coordinator *and*, *or*, or *nor*:

(34) a. Neither Mary nor Sue came early;
  *Neither Mary and Sue came early
 b. Either Mary or Sue came early;
  *Either Mary nor Sue came early
 c. Both Mary and Sue came early;
  *Both Mary nor Sue came early

We might represent these cooccurrence restrictions by treating $\langle both, and\rangle$, $\langle either, or\rangle$, and $\langle neither, nor\rangle$ as complex lexical items and code this in our representations as in (35):

(35)



Node 5 labeled *neither* precedes 6, labeled *Mary*, but node 2, which represents the complex conjunction *neither* ... *nor* ... neither precedes nor follows 6, *Mary*.

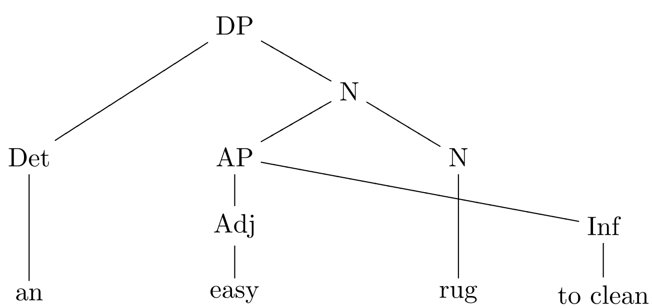
ii. **semantic units.** In mathematical languages the syntactic constituents of an expression are precisely the subexpressions which are assigned denotations. But it seems that in (36) and (37) the prenominal adjectives *easy* and *difficult* form a semantic unit with the postnominal *to*-phrase. Note that the (a,b) pairs are paraphrases of the right-hand ones in which the the adjective occurs postnominally and more clearly forms a constituent with the to-phrase.

(36) a. an easy rug to clean
b. a rug which is easy to clean

(37) a. an easy theorem to state but a difficult one to prove
b. a theorem which is easy to state but difficult to prove

Rugs of the sort mentioned in (36a) are understood to have the property expressed by *easy to clean*; (overtly expressed by the postnominal constituent *which is easy to clean* in (36b)). Typically the constituents of an expression are assigned a meaning. But *easy rug* in (36a) does not have a meaning, nor does *easy theorem* in (37b). It seems then that we want to think of *easy to clean* as having a semantic interpretation in (36a) and *easy to state* and also *difficult to prove* as having semantic interpretations in (37b). Assuming that only constituents are interpreted we can represent these judgments of interpretation by:

(38)

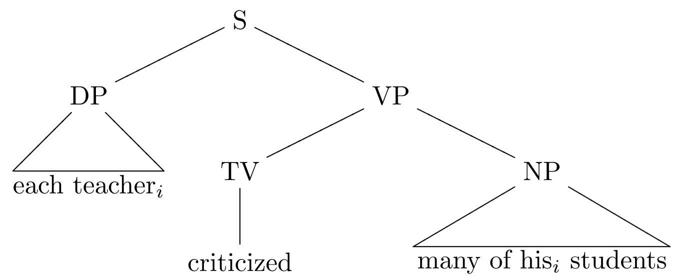


iii. **binding** (Blevins (1994)). Here we consider some expression types that play an important role in current linguistic theorizing. In expressions like (39) the pronoun *his* can be understood as bound by *each teacher*, indicated here by the use of the same subscript i.

(39) Each teacher$_i$ criticized many of his$_i$ students

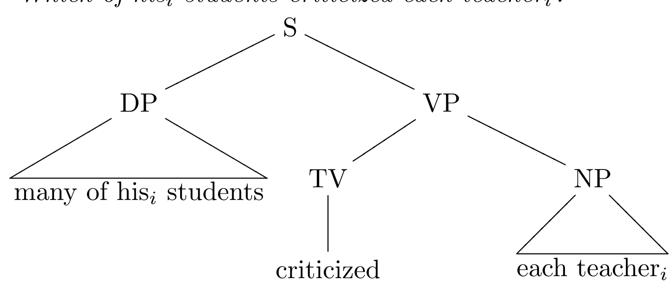
Linguists have observed that in cases like this the antecedent *each teacher* c-commands the referentially dependent expression *his* (as well as *his students* and *many of his students*). The relevant constituency relations in (39) are given by (40):

(40)



But when the c-command relations are reversed, as in (41a,b) graphed in (41c), the pronominal expressions are not naturally interpretable with *his* bound to *each teacher*.

(41) a. **Many of his$_i$ students criticized each teacher$_i$*
b. **Which of his$_i$ students criticized each teacher$_i$?*
c.



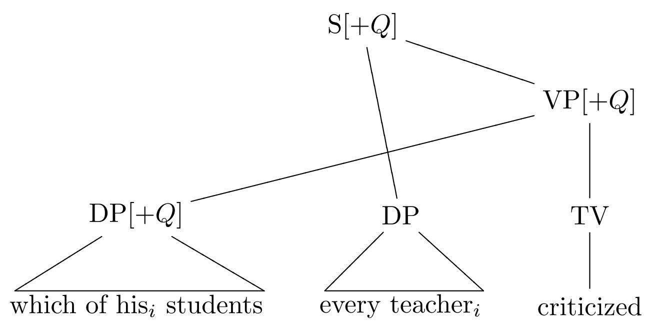
But suppose we question the object of *criticize* in (39). In such cases the interrogative DP, noted DP[$+Q$] here, occurs initially in the question, noted S[$+Q$], and the subject DP, which denotes the ones doing the criticizing, remains in place preverbally. To avoid irrelevant complications due to auxiliaries, we present the questions in an indirect context determined by the frame *I don't know* ________ .

(42) a. I don't know which of his$_i$ students each teacher$_i$ criticized.
b. I don't know how many of his$_i$ students each teacher$_i$ criticized.

Now under standard ways of presenting the constituent structure of

(42a,b) the interrogative DPs *which of his students* and *how many of his students* would not be c-commanded by *each teacher* so we should predict that we cannot interpret these Ss in such a way that the pronominal DPs are referentially dependent on *each teacher*. But in fact we can. The judgments of referential dependency are those appropriate to the case where *each teacher* c-commands the pronominal DPs. But that would be the structure on the discontinuous constituent analysis in (43). (We only graph that part of (42a,b) following *I don't know*).

(43)



Thus the discontinuous constituent (DC) analysis preserves the generalization that quantified DP antecedents of pronominal expressions c-command them.

Our purpose here is not to claim that DC analyses can be used to represent the full range of facts concerning the distribution of referentially dependent expressions and their antecedents. Much has been discovered about these relations in the past twenty years, and we have just mentioned one of the relevant facts. No current analysis adequately represents all the (known) facts. But DC analyses have not been extensively investigated in these or other regards, and we now understand that lol trees are mathematically clear and respectable objects which allow DC*s*. As students of language structure, then, we have a new tool of analysis at our disposal and should feel free to use it.

The constituent structure trees most commonly used by linguists are required to satisfy an additional conditions, called the Exclusivity Condition :

Definition 3.20. A leaf ordered tree T satisfies the Exclusivity Condition iff for all nodes b, d, if b and d are independent then $b <^* d$ or $d <^* b$. (Nodes x and y are independent, recall, iff neither dominates the other).

(31) fails Exclusivity since nodes 4 and 5 are independent but neither precedes the other. So the lol trees satisfying Exclusivity constitute a

proper subset of the lol trees.

Exercise 3.15. On the basis of the data in (a), exhibit plausible tree graphs using discontinuous constituents for the expressions in (b). (You may hide small amounts of ignorance with little triangles). State why you chose to represent the discontinuous expressions as single constituents.

a.1. More boys than girls came to the party.
Five more students than teachers signed the petition.
(Many) fewer boys than girls did well on the exam.
More than twice as many dogs as cats are on the mat.
Not as many students as teachers laughed at that joke.

a.2. *More boys as girls came to the party.
*Five more students as teachers signed the petition.
*Fewer boys as girls did well on the exam.
*More than twice as many dogs than cats are on the mat.
*Not as many students than teachers laughed at that joke.

b.1. More boys than girls
b.2. Exactly as many dogs as cats

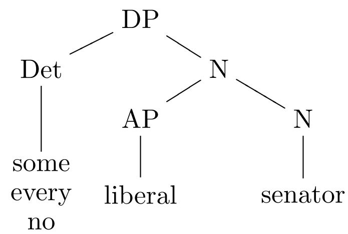
Exercise 3.16. Consider the intuitive interpretation of the Ss in (i) below:

(i.) a. some liberal senator voted for that bill
b. every liberal senator voted for that bill
c. no liberal senator voted for that bill

We can think of these three Ss as making (different) quantitative claims concerning the individuals who are liberal senators on the one hand and the individuals that voted for that bill on the other. (i.a) says that the intersection of the set of liberal senators with the set of individuals who voted for that bill is non-empty; (i.c) says that that intersection is empty; and (i.b) says that the set of liberal senators is a subset of the set of those who voted for that bill. In all cases the Adjective+Noun combination, *liberal senator*, functions to identify the set of individuals we are quantifying over (called the domain of quantification) and thus has a semantic interpretation. That interpretation does not vary with changes in the Determiner (*every*, *some*, *no*). Similarly we can replace *liberal senator* with, say, *student*, *tall student*, *tall student who John praised*, etc., without affecting the quantitative claim made by the determiners (Dets) *some*, *every*, and *no*. So the interpretation of the

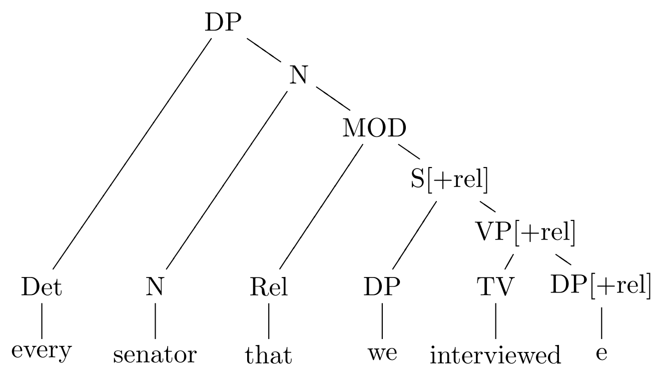
Det is independent of that of the noun or modified noun combination that follows it. These semantic judgments are reflected in the following constituent analysis:

(ii.)



Similarly, in (i.c), the relative clause *that we interviewed* functions to limit the senators under consideration to those we interviewed and thus seems to form a semantic unit with *senator* to the exclusion of the Dets *every*, *no*, ... as reflected in the constituent structure in (i.d).

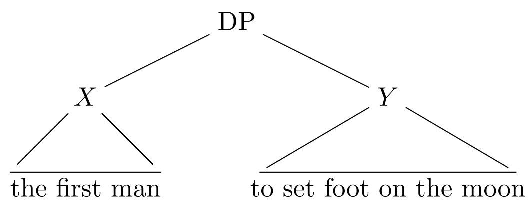
c. every senator that we interviewed; no senator that we interviewed



Current linguistic theories vary with regard to the categories assigned to the constituents in (i.b) and (i.d), but for the most part they agree with the major constituent breaks, specifically that the adjective and relative clause form a constituent with the common noun senator to the exclusion of the Dets *every*, *no*, ... But consider the expressions in (e):

e. the first man to set foot on the moon; the next village we visited; the second book written by Spooky-Pooky; the last student to leave the party

Question 1 give a semantic reason why we should not treat the apparent adjectives (*first*, *next*, ...) as forming a constituent with the following common noun (*man*, *village*, ...) to the exclusion of the material that follows the common noun. So your semantic reason should argue against a constituent structure of the sort below:



Question 2 give a semantic reason why the apparent adjectives (*first, next*, ...) should be treated as forming a unit with the expression that follows the common noun (*man, village*, ...).

Question 3 give a syntactic reason why the apparent Det the forms a syntactic unit with the apparent adjective (*first*, *next*, ...). Exhibit a discontinuous tree structure for these expression which embodies both the facts.

Exercise 3.17. Consider the DPs below:

the tallest student in the class	the fastest gun in the west
the most expensive necktie John owns	the worst movie I ever saw

a. Give a semantic reason why we do not want to treat the superlative adjective (*tallest*, *fastest*, *worst*, *most expensive*) as forming a constituent with the following common noun to the exclusion of the postnominal material (*in the class*, *in the West*, ...).

b. Give a syntactic reason why the initial occurrence of the should form a constituent with the comparative adjective to the exclusion of the common noun.

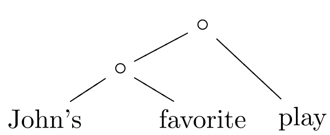
c. Exhibit a gross constituent structure for one of these DPs which incorporates these judgments. ("gross" means you can use little triangles to avoid detailing irrelevant structure).

Exercise 3.18. Consider the DPs below:

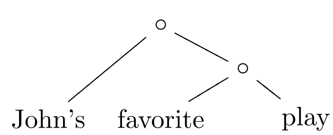
John's favorite book; his latest play; my most treasured pictures

Find reasons supporting a constituent analysis compatible with (i.a) rather than (i.b)

(a)



(b)



Exercise 3.19. Exhibit gross constituent structures for each of the Ss below Ojeda (1987).

a. *Tu quieres poder bailar tangos* "You want to be able to dance tangos"
b. *Quieres tu poder bailar tangos?* "Do you want to be able to dance tangos?"
c. *Quieres poder tu bailar tangos?* "Do you want to be able to dance tangos?"
d. *Quieres poder bailar tu tangos?* "Do you want to be able to dance tangos?"

Exercise 3.20. Provide a discontinuous tree structure for the sentence below in which *wrote down* is a constituent and *the names of my colleagues and their spouses* is a constituent. (The example is modeled on one in Chomsky (1996) pg. 324.)

I wrote the names down of my colleagues and their spouses.

Exercise 3.21. Let $T = (N, D, L, <)$ and $T' = \langle N', D', L', <' \rangle$ be lol trees. Complete the following definition correctly (the correct definition is the same regardless of whether T and T' are required to satisfy Exclusivity):

A function $h : N \to N'$ is an isomorphism from T to T' iff ________.

Exercise 3.22. Linear orders were defined in this chapter, on page 72. Let A be a finite set, say listed in some fixed order as $a_1, a_2, \ldots, a_n$. Define the *dictionary order* $\leq$ *of* A^* as follows $s \leq t$ iff there is some common prefix u of s and t such that either $u = s$, or else there is $i < j$ such that $u \frown \langle a_i \rangle$ is a prefix of s and $u \frown \langle a_j \rangle$ is prefix of t. Prove that $\leq$ is a linear order.

The dictionary order is usually called the *lexicographic order*.

Exercise 3.23. In their book on minimalist syntax, Lasnik and with Cedric Boecks (2003) define c-command as follows (p. 51):

Definition 3.21. *A c-commands B* iff (1) and (2) both hold:

a. A does not dominate B.
b. Every node that dominates A dominates B.

This is simpler than the definition which we gave in Figure 4.

Query. Are these definitions equivalent? That is, in an arbitrary tree is it the case that a node x c-commands a node y using the original definition iff x c-commands y in the alternative sense mentioned just above. Hint: the answer is NO. Give an example illustrating the difference. Consider separately the cases in which (1) the dominance relation is taken to be reflexive, and (2) the dominance relation is irreflexive.

Concluding Reflection. We have seen that order relations—ones that are transitive and antisymmetric—are basic to linguistic representation. And they will recur in various guises throughout later work in this text. It is then of some interest to consider operations on relations that preserve these order properties. Here is one basic one:

Definition 3.22. Let R be a binary relation on a set A. Then the R-*converse* of R, written R^{-1}, is the binary relation on A defined by

$$\text{for all } x, y \in A, \langle x, y\rangle \in R^{-1} \text{ iff } \langle y, x\rangle \in R.$$

For example, consider the $\leq$ relation in arithmetic. Its converse is the $\geq$ relation: $n \geq m$ iff $m \leq n$, for all numbers m, n. Equally the converse of the strictly $<$ relation is the strictly $>$ relation: $m > n$ iff $n < m$. For another example, suppose we are considering the subset relation on $\mathcal{P}(\mathbb{N})$. Its converse is the superset relation: $X \supseteq Y$ iff $Y \subseteq X$. And as well the converse of the proper subset relation, $\subset$, is the proper superset relation, $\supset$.

Now let us observe the following two properties of the converse operation—that function noted $^{-1}$ which maps a binary relation R to its converse:

Theorem 3.4. *For R a binary relation on a set A,*

a. if R is transitive then R^{-1} is transitive, and
b. if R is antisymmetric then R^{-1} is antisymmetric.

Thus we infer that if R is an order relation then so is R^{-1}.

Proof.

a. Let R be transitive. Assume that $\langle x, y\rangle \in R^{-1}$ and $\langle y, z\rangle \in R^{-1}$. We must show that $\langle x, z\rangle \in R^{-1}$. By the assumptions both $\langle y, x\rangle$ and $\langle z, y\rangle$ must be in R. So by the transitivity of R, $\langle z, x\rangle \in R$, whence by the definition of $^{-1}$, $\langle x, z\rangle \in R^{-1}$, which is what we desired to show.
b. Let R be antisymmetric. Assume $\langle x, y\rangle$ and $\langle y, x\rangle$ are both in R^{-1}. Show that $x = y$. But by the assumptions $\langle y, x\rangle$ and $\langle x, y\rangle$ are in R, so $y = x$, which is what we desired to show. □

Suppose now that we know that the converse of a relation R is transitive (antisymmetric). Can we infer that the relation R itself is transitive (antisymmetric)? The answer is yes. By the theorem immediately above we see that if a relation R^{-1} is transitive (antisymmetric) then its converse must be transitive (antisymmetric), and we see below that:

Theorem 3.5. *For R a binary relation on a set A, $(R^{-1})^{-1} = R$.*

Proof. We know that for all $x, y \in A$, $\langle x, y\rangle \in R$ iff $\langle y, x\rangle \in R^{-1}$ iff $\langle x, y\rangle \in (R^{-1}) - 1$.

Thus the pairs $\langle x, y\rangle$ in R are the same as those in $(R^{-1})^{-1}$ so the two relations are the same. $\square$

4

Syntax II: Design for a Language

Broadly speaking, a grammar $\mathcal{G}$ consists of three parts: a generative, *syntactic*, component and two interpretative components, a *phonological* one and a *semantic* one. The syntactic component defines a (typically infinite) set of expressions, the semantic component tells us what they mean, and the phonological component tells us how to pronounce them or gesturally interpret them in the case of signed languages such as ASL (American Sign Language). The *language* $\mathcal{L}(\mathcal{G})$ generated by a grammar $\mathcal{G}$ is the set of phonologically and semantically interpreted expressions it defines. A grammar $\mathcal{G}$ for an empirically given natural language L, such as English, Swahili, Japanese, etc. is said to be *sound* if all its interpreted expressions are judged by competent speakers to be expressions of $\mathcal{L}$, that is, $\mathcal{L}(\mathcal{G}) \subseteq L$. $\mathcal{G}$ is *complete* if $L \subseteq \mathcal{L}(\mathcal{G})$; that is, every expression competent speakers judge to be in L is generated by $\mathcal{G}$.

In this chapter we illustrate this by constructing a generative grammar to be called Eng. We also want to go through the process of reasoning from an existing proposal to one that is more adequate. Specifically we present a *lexicon* and some *rules* which together generate a fragment of English. In a later chapter, we illustrate how a grammar of this sort can be semantically interpreted.

We should emphasize that at the time of this writing there does not exist a sound and complete grammar for English, and extensive on-going research offers a great diversity of formats in which rules and lexicons are formulated. We attempt to be fairly generic in our approach here rather than committing ourselves to one or another particular theory. Still, explicitness requires we make some commitments if only for illustrative purposes.

4.1 Beginning Grammar

We choose as lexical items expressions we feel are not built from others. Crucial in designing the lexicon is the choice of when to assign two expressions the same grammatical category. The reason is that assigning the same category to expressions is our way of grouping them together for purposes of the rules of the grammar the ways we build complex expressions from simpler ones. Expressions with the same grammatical category are treated alike by the rules. So if a rule tells us that a string s of category C combines with a string t of category D to form a string u of category E then, in general, any other string s' of category C will combine with any t' of category D to form a string u' of category E. Choosing a category name for a lexical string is much less important than deciding that two different strings have the same category.

An issue that arises immediately in choosing a category for an expression concerns cases in which a given string apparently has more than one category. Compare the use of *walk* as a noun in (1a) and as a verb in (1b):

(1) a. We take a walk after dinner.
    b. We walk to school in the morning

In this case we feel that the verbal use of *walk* in (1b) is more basic (we do not justify this here), and that the nominal use in (1a) might reasonably be derived in some way. So rules that derive expressions from expressions have the option of changing category without audibly changing the string component of the expression. Conversely the nominal use of *shoulder* in *He hurt his shoulder* is felt to be more basic than the verbal use in *He will shoulder the burden without complaining*.

However, many apparently simple expressions have both nominal and verbal uses where we find no intuition that one is more basic than the other. Compare the nominal use of *respect* in *He gives me no respect* with the verbal use in *We respect her a lot*. Similarly *judge* and *honor* are equally easy as nouns and as verbs. But the distribution of such expressions as nouns is quite different from their use as verbs. As a noun *respect* (*judge*, *honor*) combines with possessive adjectives to form complex nominals, *my respect (for him)*, *his judge*, etc. and as a verb they form imperatives, as *in Respect your elders!*, and take past tense marking (*They respected him for his leadership qualities*).

And as an item like *respect* does not appear to be derivationally complex, it will be entered into the Lexicon for English twice: once as a noun and once as a verb. To handle these facts we represent lexical expressions, indeed expressions in general, as ordered pairs (s, C), where s is a string of vocabulary items and C is a category name. s is called the *string coordinate* of (s, C) and C is its *category coordinate*. Linguists usually write (s, C) as $[_C s]$. Thus $(honor, \mathrm{N})$ and $(honor, \mathrm{V})$ could be distinct lexical items in our grammar, ones differing just by their category coordinate. In fact we do not treat abstract nouns such as $(honor, \mathrm{N})$, but we will use extensively the possibility of assigning a given string many categories. We note too that many complex strings seem to have more than one category. For example, *Ma's home cooking* might be a Sentence, meaning the same as *Ma is home cooking*, or it might be some kind of nominal, as in *Ma's home cooking is the best there is.*

Now consider some fairly basic expressions of English which we will design our grammar Eng to generate:

(2)
a. Dana smiled.
b. Dana smiled joyfully.
c. Sasha praised Kim.
d. Kim criticized Sasha.
e. He smiled.
f. She criticized him.
g. Sasha praised Kim and smiled.
h. Some doctor cried.
i. No priest praised every senator.
j. He criticized every student's doctor.
k. Adrian said that Sasha praised Kim.

Competent speakers of English recognize (2a), ..., (2k) as expressions of English, indeed as expressions of category S (Sentence). (We shall accept the S terminology here though some theories of grammar use other category designations, such as IP "Inflection Phrase" instead of S).

Independent of the category name chosen, it is reasonable that (2a), (2b), ..., be assigned the same category. Here are three such reasons: One, each can be substituted for the others in embedded contexts like the one following *that* in (2k). Thus *Adrian said that Dana smiled* is grammatical English, as is *Adrian said that Dana smiled joyfully*, *Adrian said that Sasha praised Kim*, ... and even, *Adrian said that Dana said that Sasha praised Kim*, *Adrian said that Dana said that Robin said that ...*, etc. (see Chapter 1).[1]

Two, distinct members of this set can generally be coordinated with

[1]Intersubstitutivity as a test for sameness of grammatical category works better when applied to lexical items or expressions derived in just a few steps from lexical items than it does when applied to ones that have undergone many rule applications where various stylistic factors become more important.

and and *or* (and with a slight complication, *neither* ... *nor* ...[2]. So the expressions in (3) are grammatical and of the same category as their conjuncts - the expressions that combined with *and* and *or* in the first place.

(3) a. Either Sasha praised Kim or Kim praised Sasha.
b. Sasha praised Dana and Dana smiled.
c. Kim criticized Sasha but Adrian said that Sasha criticized Kim.
d. Either Kim praised Dana and Dana smiled or Dana praised Kim and Kim smiled.

Often expressions of the same category can be coordinated, and ones of different categories cannot. For example, Ss and NPs do not naturally coordinate:

(4) *Dana smiled joyfully or Sasha

And three, the expressions in (2) are semantically similar: all "make a claim" that is, they are true, or false, in appropriate contexts. This property relates to the traditional definition of Ss as expressions which express complete thoughts. *Dana described* cannot be said to be true or false since it is incomplete, it simply fails to make a claim. If we complete it, as in *Dana described the thief*, it then makes a claim, and (given appropriate background information) we can assess whether that claim is true or not.

So we want our grammar Eng to generate the expressions in (2) with category S. That is, $(\textit{Sasha praised Kim}, \mathrm{S})$ will be an expression in $\mathcal{L}(\mathrm{Eng})$. But these expressions are syntactically complex, so they will not be listed in $\mathrm{Lex}_{\mathrm{Eng}}$, the lexicon for Eng. Rather they will be derived by rule.

In contrast, consider the expressions *Dana*, *Sasha*, *Adrian*, *Robin*, and *Kim*. These are traditionally called *Proper Nouns* (or *Names*) and they appear to be syntactically simple and so are candidates for being lexical items. We shall in fact treat them as lexical items of category NP. That is, $(\textit{Dana}, \mathrm{NP}) \in \mathrm{Lex}_{\mathrm{Eng}}$, $(\textit{Sasha}, \mathrm{NP}) \in \mathrm{Lex}_{\mathrm{Eng}}$, etc. Note

[2]Coordination of an expression with itself is often bizarre, but not always so. The repetition in (b) below is an intensifying effect, and makes the example natural in a way in which (a) not.

a. ?Sasha criticized Kim and Sasha criticized Kim
b. Sasha laughed and laughed and laughed

We do not consider such repetition problems here. But note had we decided that (a) above were ungrammatical, we would not want to change our overall approach to coordination. Rather we would conclude that the acceptability of conjoining expressions depends on more than just having the same category.

that these expressions satisfy our criteria for being assigned the same category (regardless of what we call it). They can substitute one for another in the expressions in (2), they are semantically similar in that they all function to denote individuals, and they coordinate with each other: *both Dana and Kim*, *either both Dana and Kim or both Sasha and Adrian*, *neither Kim nor Dana*, etc. So far, then, Lex_{Eng} is a set with five elements: (*Dana*, NP), (*Sasha*, NP), etc. We abbreviate this notation slightly in giving Lex_{Eng} to date as:

(5) NP: Dana, Sasha, Adrian, Kim, Robin

Now consider how we might generate the S *Dana smiled*. The tree in (6) represents what we know so far:

(6)
```
        S
      /   \
    NP     ?

   Dana   smiled
```

We want to design a category X for *smiled* and then formulate a rule whose content will be: A string s of category NP followed by a string t of category X is a string of category S. Traditionally we might assign *smiled* the category V_i, intransitive verb. However most current theories of grammar use a more systematic notation here rather than just inventing a totally new category symbol. We shall use the notation NP\S, read as "NP under S" or "look left for an NP to become an S"[3]. The category symbol NP\S is built from other category symbols. And once we give the rules of our grammar it will follow that an expression of category NP\S is one that concatenates with a string of category NP to its left (the direction in which the "slash" \ leans) to form an expression of category S. More generally an expression of category $B\backslash A$ combines with an expression of B to its left to form an expression of category A.

So the set Cat_{Eng} of category symbols used in our grammar is not just an unstructured list, rather it is constructed from some primitive categories using some functions that build derived categories from simpler ones. Here is an initial definition with primitive members NP and S (to which we later add some others):

Definition 4.1. Cat_{Eng} is the least set satisfying (i) and (ii) below:

[3]The slash notation is taken from an approach to grammar called Categorial Grammar (see Oehrle et al. (1988)). Our use of that notation is compatible both with traditional subcategorization notation as well as current Minimalist approaches to grammar.

i. NP and S belong to $\mathrm{Cat}_{\mathrm{Eng}}$.

ii. If A and B are in $\mathrm{Cat}_{\mathrm{Eng}}$, then (A/B) and $(B\backslash A)$ are in $\mathrm{Cat}_{\mathrm{Eng}}$.

We treat the right slash, /, and the left slash, \, as two place function symbols, writing them between their arguments. We can equally well describe $\mathrm{Cat}_{\mathrm{Eng}}$ as the closure of $\{\mathrm{NP}, \mathrm{S}\}$ under the functions / and \. We might write our definition explicitly as:

(7) a. $\mathrm{Cat}_0 = \{\mathrm{NP}, \mathrm{S}\}$ and for all $n \geq 0$,

b. $\mathrm{Cat}_{n+1} = \mathrm{Cat}_n \cup \{(A/B) \mid A, B \in \mathrm{Cat}_n\}$
$\cup \{(B\backslash A) \mid A, B \in \mathrm{Cat}_n\}$.

Then $\mathrm{Cat}_{\mathrm{Eng}}$ is defined to be $\bigcup_{0<n} \mathrm{Cat}_n$. That is, a string s over the six symbols $\{\mathrm{NP}, \mathrm{S}, /, \backslash, (,)\}$ is an element of $\mathrm{Cat}_{\mathrm{Eng}}$ iff for some $n \in \mathbb{N}$, $s \in \mathrm{Cat}_n$. And the reader should be aware that there are many ways besides our "union" approach to define the closure of a set under some functions. A very common one is by intersections. Thus we might define $\mathrm{Cat}_{\mathrm{Eng}}$ to be the intersection of the sets which include $\{\mathrm{NP}, \mathrm{S}\}$ and are closed under / and \. The resulting set is the same as the one we defined via unions. Categories of the form A/B and $B\backslash A$ are called *slash categories*; B is the *denominator* category, A the *numerator*.

In general, in such cases, parentheses are needed to avoid ambiguity: $A/(B/C)$ is not the same category as $(A/B)/C$, just as $2+(3\times 4) = 14$ is different from $(2 + 3) \times 4 = 20$. (One way to avoid parentheses is to write all function symbols initially, as in *Polish notation*; another is to write them all at the end, resulting in *reverse Polish notation*. Either of these ways avoids ambiguity, but neither is as readable as writing the operation between its arguments. You will see an example of Polish notation when we turn to propositional logic in Chapter 7, see page 7.2.3.) That said, we eliminate parentheses from category names when no confusion results.

Having defined the category symbols we will use we now define our first structure building operation. It is a binary function called FA (Function Application) defined by:

(8) For all $A, B \in \mathrm{Cat}_{\mathrm{Eng}}$,

a. $\mathrm{FA}((s, B), (t, A/B)) = (t \frown s, A)$, and

b. $\mathrm{FA}((s, B), (t, B\backslash A)) = (s \frown t, A)$.

We understand here that the domain of FA is the set of all pairs of possible expressions consisting of a string s of any category B and a string t either of category A/B or of category $B\backslash A$, A any category. Informally we can read this as saying that a t of category A/B is looking to the right for any s of category B, and it will concatenate with s to form $t \frown s$ of category A. Similarly a t of category $B\backslash A$ is looking left for a string s of category B and s will concatenate with t to form an expression $s \frown t$ of category A. So t is always looking in the direction the slash leans.

Note that there are no conditions on the rule FA, but normally in defining a structure building function for a grammar, we stipulate a variety of Conditions on the domain of the function. These conditions limit what the function applies to—what it can "see". These constraints are actually responsible for much of the "structure" a particular grammar has Keenan and Stabler (2003). Later on, functions we discuss will have a more specialized role in the grammar and only apply to expressions that satisfy various conditions, both on their category and on their string coordinates.

We now enrich the lexicon of Eng by adding:

(9) NP\S: smiled, laughed, cried, grinned

The criteria we have been using support treating these as lexical expressions of the same category. They can substitute for each other, as in: *Dana said that Kim smiled* $\Longrightarrow$ *Dana said that Kim laughed*, etc. They all denote activities that individuals may experience, and they coordinate among themselves: *Kim both laughed and cried*, *Dana neither laughed nor cried*, etc. And with this category assignment we can generate (*Dana smiled*, S) by applying FA to the relevant lexical items:

(10) $\mathrm{FA}((\textit{Dana}, \mathrm{NP}), (\textit{smiled}, \mathrm{NP}\backslash\mathrm{S})) = (\textit{Dana smiled}, \mathrm{S})$.

We now have a small lexicon and a set of Rules {FA}. So $\mathcal{L}$(Eng), the language generated from the lexicon by the rules, is small but non-empty. Recall:

Definition 4.2. $\mathcal{L}$(Eng) is the least set of possible expressions which includes $\mathrm{Lex}_{\mathrm{Eng}}$ and is closed under the rule FA.

Here is a natural way to represent the argument that (*Dana smiled*, S) is an expression of $\mathcal{L}$(Eng).

(11) (Dana smiled, S)

FA

(Dana,NP) (smiled,NP\S)

We refer to such tree structures as *Function-Argument (F–A) trees*. The leaves of F–A trees are lexical items. At each mother node in the tree, we indicate the function which applied to the daughters and what the value of that function is at the daughters. Typically, just one function could apply; that is, the daughters only lie in the domain of one of the structure building functions. Thus we can omit the designation of the function without loss of information. The sense in which (11) represents the argument that (*Dana smiled*, S) is in $\mathcal{L}$(Eng) is as follows: We know that the leaves of the tree are in $\mathcal{L}$(Eng) since they are lexical items. And we know that $\mathcal{L}$(Eng) is closed under the structure building functions RFA and LFA. Hence

$$\text{LFA}((\textit{Dana}, \text{NP}), (\textit{smiled}, \text{NP}\backslash\text{S})) = (\textit{Dana smiled}, \text{S})$$

is in $\mathcal{L}$(Eng).

But note that our generating function FA is concatenative, so we can represent derivations of expressions by standard trees as well.

(12)

$$\begin{array}{ccc} & \text{S} & \\ \text{NP} & & \text{NP}\backslash\text{S} \\ | & & | \\ \text{Dana} & & \text{smiled} \end{array}$$

The simple category forming apparatus we have at hand allows us to form categories quite productively for a variety of expressions not yet considered. Consider for example:

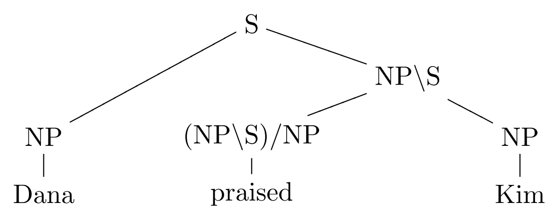
(13) Dana praised Kim

We want this to be a string of category S for reasons given earlier (substitution with other Ss, coordination with them, meaning similarity), and we are already committed to assigning *Dana* and *Kim* the category NP. At issue is to find a category for *praised* which permits the sequence of lexical items to cancel to S. There are a couple of logically acceptable candidates, but the best one draws on the sort of linguistic reasoning we have been using. Namely (14) shows that *praised Kim* coordinates with lexical expressions of category NP\S, suggesting that it should have that category:

(14) a. Dana both praised Kim and smiled.
b. Dana neither praised Kim nor smiled.

And if we treat *praised Kim* of category NP\S, it suffices to assign *praised* the category (NP\S)/NP. For then by FA, it will combine with the NP *Kim* on its right to form an NP\S, *praised Kim*. You will find the standard tree summarizing this discussion in (15) below; the F–A tree is similar.

(15)



So let us further enrich the Lexicon of Eng by:

(16) (NP\S)/NP: praised, criticized, interviewed, teased

Note that such expressions coordinate easily:

(17) a. Dana both praised and criticized Kim.
b. Dana neither praised nor criticized Kim.

As well they are often intersubstitutable in embedded contexts, and they are semantically similar in expressing a binary relation between individuals, such as Dana and Kim above.

The relation is expressed in two steps: Dana is in the *praise* relation to Kim iff Dana has the property expressed by *praised Kim*.

4.2 Manner Adverbs

Let us move on to manner adverbs such as *joyfully* in (2b). In general, the result of combining a manner adverb with an NP\S yields an NP\S that is meaningful in the same way as the original one. Indeed observe the entailment below:

(18) a. Dana smiled joyfully.
b. Dana smiled.

To say that a sentence P *entails* a sentence Q, noted $P \models Q$, is just to say that Q is interpreted as True in every situation (model) in which P is True. So the truth of P guarantees that of Q. See Section 7.1 for a discussion of entailment.

Let us now add manner adverbs such as *joyfully*, *quickly*, and *carefully* to Lex_{Eng}, as in ((2b). Such modifiers combine with expressions of category NP\S to form ones that coordinate with expressions in that same category, as in *Dana smiled joyfully and praised Kim.* So they should combine with an NP\S on the left to form an NP\S. This means that they should have category

(NP\S)\(NP\S)

as in the F–A tree (19a) or the standard tree (19b). In (19a) we have not explicitly noted the two uses of FA which derived the non-lexical items.

(19) a. (Dana smiled joyfully,S)

(smiled joyfully,NP\S)

(Dana,NP) (smiled,NP\S) (joyfully,(NP\S)\(NP\S))

b. S

NP\S

NP NP\S (NP\S)\(NP\S)

Dana smiled joyfully

Formally we enrich LexEng to include:

(20) (NP\S)\(NP\S): joyfully, quickly, carefully, tactfully

Simplifying notation We shall often write P_1, "one place predicate", for NP\S. Similarly, P_2, "two place predicate", abbreviates

$$(NP\backslash S)/NP$$

(and P_3, "three place predicate" abbreviates

$$((NP\backslash S)/NP)/NP).$$

Using this notation, manner adverbs have category $P_1\backslash P_1$. Occasionally we will use P_0, "zero place predicate", instead of S. For $n \geq 0$, P_ns are n place predicates, and hence combine with n NP arguments to form a P_0; the order of combination is rightward, except that the last NP is to the left.

Pronouns At first blush it might seem that *he* and *she* could be added to the Lexicon in the category NP, the same as for proper nouns like *Dana* and *Kim*. They can, for example, grammatically replace *Dana* in *Dana laughed*, they may coordinate with NPs, as in *Both he and Kim praised Robin*. And they seem semantically similar in that in *He laughed*, we understand that *he* refers to an individual, just as in the case of *Robin laughed*.

Despite these similarities, however, these pronouns have a vastly different distribution from proper nouns. For example, *he* (*she*) cannot grammatically replace *Robin* in (21a), which would yield the ungrammatical (21b,c).

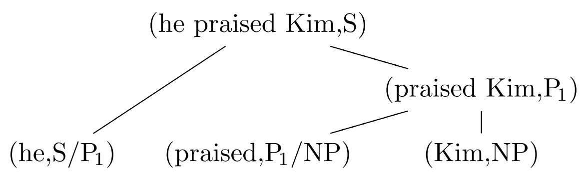
(21) a. Kim praised Robin.
b. *Kim praised he.
c. *Kim praised she.

Traditionally, *he* and *she* are *nominative* pronouns, combining with P_1s to form a P_0 (Sentence). We shall refer to such occurrences of NPs as *subject* occurrences. So *Dana* is the subject of *Dana praised Kim*. (We also call it the subject of the P_1 *praised Kim* and also the subject of the P_2 *praised*). To continue the traditional account, *he* and *she* are subject pronouns, but do not occur as objects NP occurrences which combine with a P_2 to form a P_1 or a P_3 to form a P_2. We may capture the relevant distributional facts for subjects by the following category assignment to *he* and *she*:

(22) $S/(\mathrm{NP}\backslash \mathrm{S})$: he, she, they

Thus we generate *He praised Kim* as per (23):

(23)



But our grammar will not generate (*Robin praised he*, S) since *praised* has category P_1/NP and so is looking right for an NP to make a P_1. But *he* does not have category NP. Similarly *he* does not have a left looking category $X\backslash Y$, so FA cannot apply.

Exercise 4.1. Add *him*, *her*, and *them* to the Lexicon, assigning them the same category, in such a way that the grammar generates *Kim praised him* but **Him laughed*. Say in words why the grammar does not generate (*Him laughed*, S).

This simple addition of pronouns with their restricted distribution turns out to have some unexpected consequences for the category assignment to proper nouns. We have been taking coordination as a guideline for sameness of category. But given the grammaticality of (24a,b) below, this argues that *Kim* should have the same category as *he* and also the same category as *him*, a contradiction as these two categories must be different!

(24) a. Both he and Kim laughed joyfully.
b. Dana praised both him and Kim.

This is really our first interesting problem in category assignment. Here we offer a brute force solution whose main merit is empirical adequacy. But first we must enter coordinations in the grammar, since that is the environment which triggers our problem.

Coordination Traditional categorial grammar would attempt to extend Eng so that it generates *(both) Kim and Dana*, *neither laughed nor* *cried*, *either praised Robin or criticized Adrian* by assigning an appropriate category to *and*, *or*, and *nor*. But this approach is not without problems. One of some interest is that it forces a coordination such as *Kim and Dana* to have an internal constituent structure, typically

[Kim [and Dana]],

where *Dana* is subordinate to *Kim*, specifically it is c-commanded by it.

We opt for a different approach. Linguists have observed that coordinate structures differ from subordinate ones, ones constructed with conjunctions such as *because*, *since*, etc., in that certain processes must treat each conjunct alike, whereas only the main clause is affected by that process in a subordination context. An example here is the "across the board" constraint on relativization. Consider the coordination in (25a). We can simultaneously relativize into each conjunct, as in (25b), but we cannot relativize into just one of the conjuncts, (25c,d).

(25)
a. Sasha praised Kim and Adrian criticized Dana.
b. the woman who$_i$ Sasha praised t_i and Adrian criticized t_i
c. *the woman who$_i$ Sasha praised Kim and Adrian criticized t_i
d. *the woman who$_i$ Sasha praised t_i and Adrian criticized Kim

In contrast, in Ss built with subordinate clauses, we can just relativize into the main clause, (26a) below, not the subordinate one, (26b) or both, (26c):

(26)
a. the woman who$_i$ Sasha praised t_i because Adrian criticized Kim.
b. *the woman who$_i$ Sasha praised Kim because Adrian criticized t_i.
c. *the woman who$_i$ Sasha praised t_i because Adrian criticized t_i.

So we will present coordination rules independently of Function Application. Once their behavior is well understood, perhaps they can be insightfully assimilated to the slash notation. We begin by adding a new primitive category to $\mathrm{Cat}_{\mathrm{Eng}}$, namely Conj (coordinate Conjunction). The lexicon is extended by

(27) Conj: and, or, nor

Then we give the coordination Rule, Coord in Figure 5. In giving examples we may ease readability by omitting the use of *both* and *either*. (We could extend the Coord rule to allow this formally but our interest

$$(c, \mathrm{Conj})(s, C)(t, C) \Longrightarrow \begin{cases} (\text{both } s \text{ and } t,\, C) & \text{if } c = \text{and} \\ (\text{either } s \text{ or } t,\, C) & \text{if } c = \text{or} \\ (\text{neither } s \text{ nor } t,\, C) & \text{if } c = \text{nor} \end{cases}$$

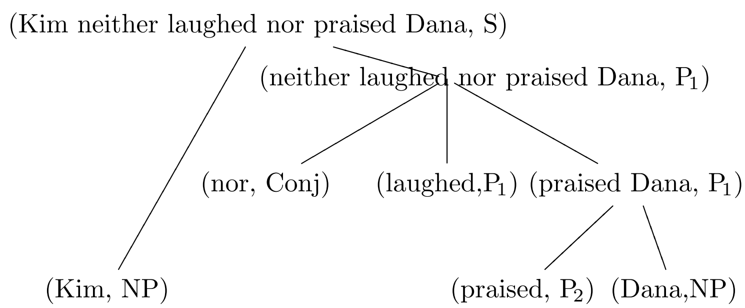
C must be one of P_0, P_1, P_2, $\mathrm{P}_1\backslash\mathrm{P}_1$, $\mathrm{P}_0/\mathrm{P}_1$ or $\mathrm{P}_2\backslash\mathrm{P}_1$.

FIGURE 5 The coordination rule, Coord.

here does not lie in the optionality of these items).

You should check that $\mathcal{L}$(Eng) is infinite. Here is one of the new expressions in it, along with its F–A tree:

(28)



Exercise 4.2. Provide F–A trees for each of the following

a. Dana either praised or criticized Sasha.
b. Dana criticized Sasha neither joyfully nor tactfully.
c. Both Dana cried and Sasha laughed.

Let us turn now to the problematic cases involving coordinations of pronouns and proper nouns. Clearly we want our grammar to generate (29) as an S:

(29) Either he or Sasha criticized Kim.

Currently, Eng does not generate (29), since it only coordinates expressions of the same category, and *he* and *Sasha* have different categories: S/(NP\S) and NP, respectively. In fact, NP is not among the coordinable categories, so at the moment Eng will not even generate *Either Adrian or Sasha criticized Kim.*

We overcome both shortcomings by allowing proper nouns to have the category S/(NP\S), as well as their current category, NP. Specifically, enrich $\mathrm{Lex}_{\mathrm{Eng}}$ by:

(30) a. $\mathrm{P}_0/\mathrm{P}_1$: Sasha, Adrian, Dana, Kim, Robin
b. $\mathrm{P}_2\backslash\mathrm{P}_1$: Sasha, Adrian, Dana, Kim, Robin

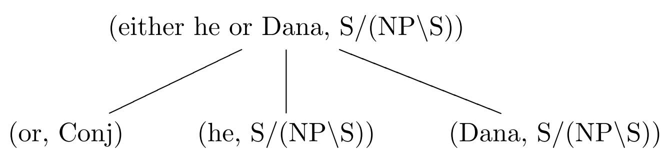
Thus proper nouns combine with P_1s on the right to form a P_0, and they combine with P_2s on the left to form a P_1. (And in general, proper nouns will combine with P_{n+1}s to form P_ns). To anticipate a worry, we note that

(31) $(\mathit{Kim}, \mathrm{NP})$, $(\mathit{Kim}, (\mathrm{P}_0/\mathrm{P}_1))$, $(\mathit{Kim}, (\mathrm{P}_2\backslash\mathrm{P}_1))$, and $(\mathit{Kim}, (\mathrm{P}_3\backslash\mathrm{P}_2))$

are four different lexical expressions (with the same string coordinate but different category coordinates). So they can be assigned different denotations by a semantic interpretation function.

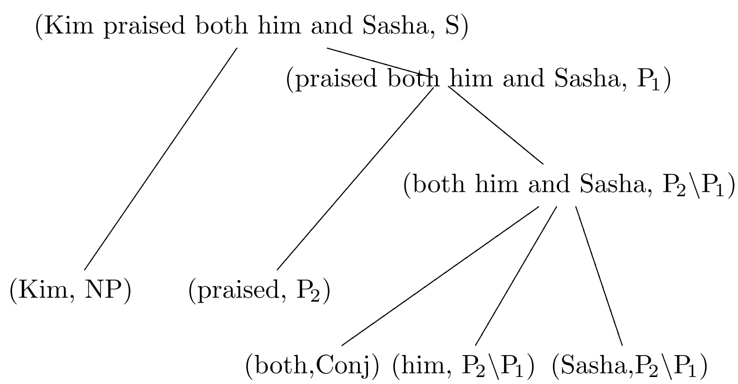
Here now are some F–A trees illustrating coordination with proper nouns, which can be coordinated in category $\mathrm{P}_0/\mathrm{P}_1$ or $\mathrm{P}_2\backslash\mathrm{P}_1$.

(32)



From this we can build the S *either he or Dana smiled.*

(33)



Note that the result of replacing $(\mathit{Kim}, \mathrm{NP})$ here with $(\mathit{Kim}, \mathrm{P}_0/\mathrm{P}_1)$ also yields a good derivation, but with different categories. As the sentence is not felt to be semantically ambiguous, when we give a semantics for this language, we must make sure that the two structures are in fact interpreted the same.

Exercise 4.3. Provide F–A trees for the following, recalling that NP is still not coordinable.

a. Either Dana or Kim criticized Sasha
b. Either both Dana and Kim or both Dana and Adrian criticized Sasha

Exercise 4.4. For each of the Ss below, provide a syntactic analysis tree. You must invent a category for *at*. Give a few reasons to support your analysis.

a. Robin smiled joyfully at Sasha
b. Robin smiled at Sasha joyfully

Quantified NPs We want to extend Eng so that it generates expressions such as *some doctor*, *no priest*, *every senator*, etc. as they occur in Ss like (2h) and (2i). Since these expressions combine with P_{n+1}s to form P_ns quite generally, they appear to have the same distribution as proper nouns and so shall be assigned some of the same categories. But they also have some internal structure, consisting of a Det (*every*, *no*, *some*, etc.) and a common N (*doctor*, *priest*, *lawyer*, etc.). Common nouns exhibit a few similarities to P_1s, but there are also very many differences. For example, P_1s are marked for tense (present *walks*, past *walked*) and person (*I walk*, *she walks*) whereas Ns are not. So again we shall take the safe if unimaginative route and treat N (Noun) as a new primitive category. Thus we enrich the Lexicon as follows:

(34) N: doctor, lawyer, priest, student, teacher
$(\mathrm{P}_0/\mathrm{P}_1)/\mathrm{N}$: every, no, some, the, a
$(\mathrm{P}_2\backslash\mathrm{P}_1)/\mathrm{N}$: every, no, some, the, a

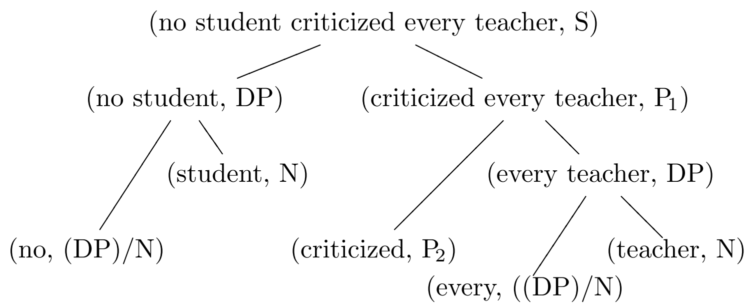
Clearly Dets like *every*, *some*, *no*, *most*, etc. combine with Ns on the right to yield "DPs" (Determiner Phrases)— expressions that combine appropriately with P_{n+1}s to from P_ns. So in one way or another we want to say that *every student* behaves as though it had category $\mathrm{P}_0 \rightarrow \mathrm{P}_1$, since it combines with P_1 on the right to form P_0s, as in *Every student laughed*. But equally it combines with P_2s on the left to form P_1s, as in *John praised every student* and finally it combines with P_3s on the left to form P_2s, as in *John gave every student a book*. So let us say that a sequence of slash categories is *unifiable* iff their denominator categories are all distinct. So for $C = \langle C_1, \ldots, C_n \rangle$ a sequence of unifiable slash categories each C_i is of the form A_i/B_i or $B_i\backslash A_i$. And we will say that a string s has category C iff for each $1 \leq i \leq n$, it combines with expressions of category B_i appropriately to form expressions of category A_i. (By "appropriately" here we just mean that s finds its B_i on the side towards which the slash in category C_i leans).

Specifically now, we use DP for the category $\langle P_0/P_1, P_2\backslash P_1, P_3\backslash P_2\rangle$. We enrich the Lexicon as follows (note that *Sasha*, *Kim*, etc. still have category NP):

(35) DP: Sasha, Adrian, Kim, Dana, Robin
DP/N: every, some, no, most, the, a

We can now generate *No student criticized every teacher*, as shown by the F–A tree below:

(36)



Exercise 4.5. Provide F–A trees for each of the following:

a. Kim and some student laughed.
b. Sasha interviewed some student and every teacher.
c. They interviewed either him or the teacher.
d. Neither Kim nor Adrian criticized every teacher.

Exercise 4.6. Design a category for *give* and *offer* below, and then exhibit the F–A trees for (a) and (b). Assume that *apple* is a N.

a. Every student gave some teacher an apple.
b. No student offered every teacher an apple.

Exercise 4.7. Add traditional adjectives such as *tall*, *industrious*, *clever*, and *female* to the Lexicon. Exhibit F–A trees for the following, using the category you have found.

a. No clever student praised every industrious doctor
b. Every industrious female student laughed

Possessives In English, possessors such as *Kim's* in *Kim's doctor* behave like Dets in the sense of occurring in the same prenominal position: **Every Kim's doctor*, etc. So we should like to add *'s* to our Lexicon in such a way that it combines with a DP on the left to form a Det. Here is one solution:

(37) DP\(DP/N): 's.

Here is a simple sentence built with *'s*:

(38)

- (some doctor criticized Dana's teacher, P_0)
  - (some doctor, DP)
    - (some, (DP)/N)
    - (doctor, N)
  - (criticized Dana's teacher, P_1)
    - (criticized, P_2)
    - (Dana's teacher, DP)
      - (Dana's,(DP)/N)
        - (Dana, DP)
        - ('s,((DP)\((DP)/N))
      - (teacher, N)

Exercise 4.8. Exhibit F–A trees for the expressions below

a. Every student's doctor laughed.
b. Kim's teacher's doctor cried.

Sentence Complements We are concerned here with Ss built from verbs like *think*, *say*, and *believe*. These combine with Sentence Complements to form a P_1, as in (39).

(39) Adrian said that Sasha praised Kim

The sentence complement consists of a full S preceded by *that*, called a *complementizer*. Linguists distinguish the category of *Sasha praised Kim*, which is S (Sentence), from that of *that Sasha praised Kim*, called a CP "Complementizer Phrase". Here, too, we shall make a category distinction, though instead of the cumbersome 'Complementizer Phrase' we shall use $\overline{\mathrm{S}}$ (read: "S bar"). One reason for making a category distinction here is that sometimes the presence vs. absence of the complementizer leads to differences in interpretation. Compare (40a,b).

(40) a. Kim believes either that there is life on Mars or that there isn't.
b. Kim believes that either there is life on Mars or there isn't.

In (40a), we have a disjunction of two $\overline{\mathrm{S}}$s. The whole S claims that Kim believes one of the disjuncts, though the speaker is not sure which. In contrast, (40b) seems to simply assert that Kim believes a certain disjunction So (40a,b) differ in meaning, and they differ in form just by the presence vs absence of the complementizer *that*.

Accepting this category distinction then we are obliged to add a new primitive category to our grammar, $\overline{\mathrm{S}}$. And we enter *that* into the

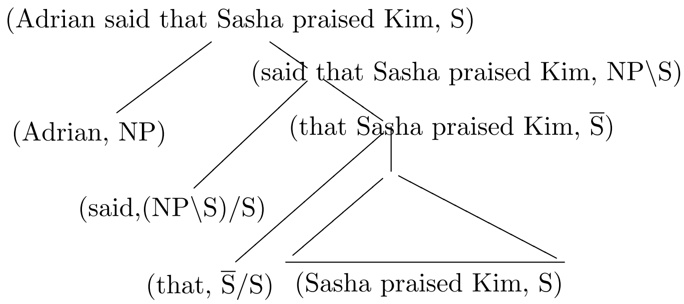
Lexicon as in (41), and sentence complement taking verbs as in (42).

(41) $\overline{\mathrm{S}}$/S: that

(42) (NP\S)/$\overline{\mathrm{S}}$: think, say, believe, regret, resent

Then (43) is an F–A tree for (39), in which the derivation of the embedded S *Sasha praised Kim* is omitted as familiar.

(43)



Let us note a last somewhat subtle distinction to be made among the sentence complement taking verbs. For many such verbs, especially semantically "weak" ones like *say*, *think*, and *believe*, the use of the complementizer *that* is optional.

(44) a. Adrian said (that) Sasha was fleeing the country.
b. Sasha thought Adrian said that Kim criticized Robin.

But for some "semantically richer" verbs, the complementizer is not easily omitted:

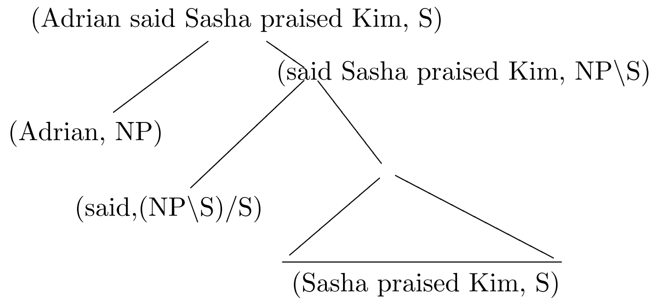
(45) a. Winston resented that his wife was wealthier than him.
b. ??Winston resented his wife was wealthier than him

Though judgments are not always crisp, let us accept as a first approximation to reality that some sentence-complement-taking verbs require a complementizer, and for others it is optional. This regularity can then be captured by allowing a second categorization in the Lexicon of the optional-that verbs:

(46) (NP\S)/S: say, think, believe

Thus our grammar also generates (47).

(47)



Note that in (47) *said* cannot be replaced by *resented* as the latter does not combine with S to form anything. *resented* only combines with $\overline{\mathrm{S}}$, and no substring of (47) is an $\overline{\mathrm{S}}$.

Exercise 4.9. Exhibit an F–A tree for each S below. Describe a situation in which (a) is true and (b) is not.

a. Sasha believes either that Kim laughed or that Dana laughed.
b. Sasha believes that either Kim laughed or Dana laughed.

Summary Grammar For purposes of later reference, we summarize in Figure 6 our grammar Eng as developed so far. We use the category abbreviations where convenient.

Remarks on Eng $\mathcal{L}$(Eng) contains several fundamental structure types in natural language: Predicate+Argument expressions, Modifier expressions, Sentence Complements, Possessives, and coordinations. Arguably all languages have expression types of these sorts. The reader might get the impression that we could attain something like a sound and complete grammar for English just by continuing in the spirit in which we have already been moving. But this would be naive. There are simply a great number of linguistic phenomena we have not attempted to account for: Agreement phenomena, impersonal constructions, extraposition phenomena, clitics, selectional restrictions, Raising, nominalizations, ellipsis, ... The structure types we have considered are all built by concatenating expressions with great freedom beginning with lexical items. But natural languages present a significant variety of expression types which generative grammarians have treated with different types of structure building operations, specifically movement operations. Here we consider one such basic case, Relative Clauses. We extend Eng to account for these structures, together with various constraints to which they are subject.

Cat_{Eng} is the closure of $\{\text{Conj}, \text{N}, \text{NP}, \text{S}, \overline{\text{S}}\}$ under / and \.
(DP is used as a variable ranging over P_1/P_0, $\text{P}_2\backslash\text{P}_1$, and $\text{P}_3\backslash\text{P}_2$.)

Categories of Lex_{Eng} are listed below, with vocabulary items:
N: doctor, lawyer, student, teacher
NP\S: smiled, laughed, cried, grinned
(NP\S)/NP: praised, criticized, interviewed, teased, is
(NP\S)/$\overline{\text{S}}$: think, say, believe, regret, resent
(NP\S)/S: say, think, believe
(NP\S)\(NP\S): joyfully, quickly, carefully, tactfully
NP: Dana, Sasha, Adrian, Kim, Robin
DP: Dana, Sasha, Adrian, Kim, Robin
(DP)/N: every, some, no, the, a
P_0/P_1: he, she, they
$\text{P}_2\backslash\text{P}_1$: him, her, them
N/N: tall, industrious, clever, female
$(\text{P}_1\backslash\text{P}_1)/(\text{P}_0/\text{P}_1)$: at, to
Conj: and, or, nor
DP\(DP/N): 's
$\overline{\text{S}}$/S: that

The rules are listed below:

a. FA (Function Application):
   For all $A, B \in \text{Cat}_{\text{Eng}}$, for all strings s, t of vocabulary items,
   a. $(s, \text{A}/\text{B}), (t, \text{B}) \Longrightarrow (s\frown t, \text{A})$.
   b. $(t, \text{B}), (s, \text{A}\backslash\text{B}) \Longrightarrow (t\frown s, \text{A})$.
   There are no conditions associated with RFA and LFA.
b. Coord (coordination)

$$(c, \text{Conj})(s, C)(t, C) \Longrightarrow \begin{cases} (\text{both } s \text{ and } t,\ C) & \text{if } c = \text{and} \\ (\text{either } s \text{ or } t,\ C) & \text{if } c = \text{or} \\ (\text{neither } s \text{ nor } t,\ C) & \text{if } c = \text{nor} \end{cases}$$

   C must be one of P_0, P_1, P_2, $\text{P}_1\backslash\text{P}_1$, or DP.

FIGURE 6 Our grammar Eng up until this point.

4.3 Relative Clauses

Consider the DP in (48a). The substring following the N *teacher* is called a *relative clause*. Most approaches to generative grammar would derive it from something like (48b) by moving the wh- word (*who* in this case) to the complementizer position in front of the S. That position is here filled by a special expression e which is not phonologically interpreted. The position from which the wh- word was moved is filled by another unpronounced symbol, t, called a *trace*. The trace and the moved wh- word are co-indexed, enabling one to retrieve the site from which movement took place.

(48) a. every teacher who$_i$ Kim criticized t_i
b. every teacher [e [Kim criticized who]]

Linguists have discovered quite a variety of constraints regulating the formation of relative clauses (RCs) as well as other syntactically related phenomena such as wh- questions as in (*Who$_i$ did Kim criticize* t_i*?*). On this standard approach, these constraints are given as constraints on the positions from which the wh- word can move and the material across which it is moved. Below we summarize instances of these constraints. Then we formulate RC Formation within the format we have been presenting, and we show how the classical constraints are satisfied.

Some classical constraints on Relative Clause Formation (RCF)

a. No Vacuous Binding. The remnant following the *wh*-word must contain an appropriate gap (traditionally marked t): **every teacher who Kim criticized Sasha*.
b. The Coordinate Structure Constraint. We cannot relativize into just one conjunct of a coordinate expression. So we have **every teacher who$_i$ Kim criticized Robin and Adrian praised* t_i But a systematic phenomenon called the Across the Board exception allows relativization into all conjuncts simultaneously, as in *Every teacher who$_i$ Kim criticized* t_i *and Robin praised* t_i. We discussed this above in (25) and (26).
c. Subjacency. Given a sentence with an RC, one cannot relativize out of that RC: So we have *John knows the student who$_i$ [* t_i *criticized the teacher]* but not **I see the teacher who$_j$ [John knows the student who$_i$ [* t_i *criticized* t_j*]]*.
d. The Empty Category Principle (ECP): We can relativize the subject of a sentence complement provided it is not preceded by a complementizer. Here are some relevant examples:

(49) a. John said the teacher criticized Amy
b. the teacher who$_i$ John said t_i criticized Amy
c. John said that the teacher criticized Amy
d. *the teacher who$_i$ John said that t_i criticized Amy

e. Pied Piping: we can relativize possessors provided the entire possessive DP is incorporated into the *wh-* word. For example, *John knows every student's teacher* and also *every student [[whose teacher]$_i$ John knows t_i]* but not **every student who$_i$ John knows [t_i 's teacher]*

f. *Wh-* phrases coordinate, but not lexical ones. So we have

the student [[whose teacher and whose doctor]$_i$ Dana interviewed t_i]

but not

**the student [[who and whose teacher]i Dana interviewed t_i]*

Similarly, conjuncts into which we have relativized cannot be exhausted by the relativization site, as in the starred example below. The first example (not generated by Eng) shows that in carefully selected cases it is possible in English to relativize into coordinate DPs.

(50) a. the senator who Sue interviewed several friends of t and several enemies of t

b. *the senator who Sue interviewed t and several enemies of t

The first five constraints are representative of major ones the standard analysis is subject to. The sixth is less widely acknowledged but in fact difficult to capture on the standard analysis.

Let us now extend Eng to a grammar Eng* in Figure 7 whose language is a superset of $\mathcal{L}$(Eng) which includes RC*s* and satisfies the constraints above.

The core idea. Wh- words, such as *who* and *whose teacher* will combine with expressions which resemble Ss, but which lack a NP argument. This gap will be coded in the category of what follows the wh-words. In the simplified version of RCF we present here, we allow only NP gaps; in richer versions one would allow Prepositional Phrase gaps as well, as in *the man [with whom]$_i$ Mary went to the movies t_i* (see Keenan and Stabler (2003)), So we will now allow categories of the form S[NP], meaning an S with an NP gap. Similarly (NP\S)[NP] is a P_1 with an NP gap, etc.

RC*s* themselves have category N\N, they combine with a N to their left to form a N. The main point on our treatment is the following: In forming DPs with RC*s*, only rules of FA, with or without feature passing are used. So we have *no movement rules* per se.

We need some new rules of function application. The idea in the FA[NP] rule is that each instance of FA in Eng now extends so that one but not both of the two items concatenated may carry the feature [NP], which is passed to the category of the derived expression. We impose one limitation on the pairs in the domain of FA[NP]. This limitation is a response the constraints on English relative clauses that we noted above.

$$\mathrm{Cat}^*_{\mathrm{Eng}} = \mathrm{Cat}_{\mathrm{Eng}} \quad \cup \{C[\mathrm{NP}] \mid C \in \mathrm{Cat}_{\mathrm{Eng}}\}$$
$$\cup \{(\mathrm{N}\backslash\mathrm{N})/(\mathrm{S}[\mathrm{NP}]), ((\mathrm{N}\backslash\mathrm{N})/(\mathrm{S}[\mathrm{NP}]))/\mathrm{N}\}$$

$\mathrm{Lex}^*_{\mathrm{Eng}}$ includes $\mathrm{Lex}_{\mathrm{Eng}}$ plus the following special items:

$\mathrm{NP}[\mathrm{NP}] : t$
$(\mathrm{N}\backslash\mathrm{N})/(\mathrm{S}[\mathrm{NP}])$: who, that
$((\mathrm{N}\backslash\mathrm{N})/\mathrm{S}[\mathrm{NP}])/N$: whose

Rule Eng* adds two rules to those of Eng: "feature passing" extensions of those FA rules, called FA[NP], and one additional clause on the Coord rule.

rule name	how it works	conditions
RFA[NP]	$(s, A/B[\mathrm{NP}]), (t, B) \Longrightarrow (s \frown t, A[\mathrm{NP}])$	see below
	$(s, A/B), (t, B[\mathrm{NP}]) \Longrightarrow (s \frown t, A[\mathrm{NP}])$	
LFA[NP]	$(t, B[\mathrm{NP}]), (s, B\backslash A) \Longrightarrow (t \frown s, A[\mathrm{NP}])$	none
	$(t, B), (s, (B\backslash A)[\mathrm{NP}]) \Longrightarrow (t \frown s, A[\mathrm{NP}])$	

In order to apply RFA[NP] to $(s, \overline{\mathrm{S}}/\mathrm{S})$ and $(t, \mathrm{S}[\mathrm{NP}])$, we require that none of the immediate constituents of $(t, \mathrm{S}[\mathrm{NP}])$ be of the form $(u, \mathrm{NP}\backslash\mathrm{S})$.

The additional clause in the Coord rule is:

$$(c, \mathrm{Conj})(s, C)(t, C) \Longrightarrow \begin{cases} (\text{both } s \text{ and } t,\ \mathrm{C}[\mathrm{NP}]) & \text{if } c = \text{and} \\ (\text{either } s \text{ or } t,\ \mathrm{C}[\mathrm{NP}])) & \text{if } c = \text{or} \\ (\text{neither } s \text{ nor } t,\ \mathrm{C}[\mathrm{NP}])) & \text{if } c = \text{nor} \end{cases}$$

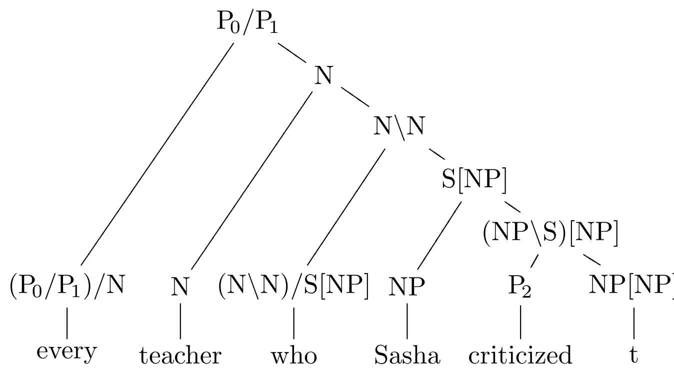
C must be one of P_0, P_1, P_2, $\mathrm{P}_1\backslash\mathrm{P}_1$, $\mathrm{P}_0/\mathrm{P}_1$ N/N, N\N, or (N\N)/(S[NP]), or some category C[NP], where C is P_0, P_1, P_2, $\mathrm{P}_0/\mathrm{P}_1$, $\mathrm{P}_2\backslash\mathrm{P}_1$, or $\mathrm{P}_1\backslash\mathrm{P}_1$.
Also we require that nether (s, C) nor (t, C) belongs to

$$\{(\textit{who}, (\mathrm{N}\backslash\mathrm{N})/\mathrm{S}[\mathrm{NP}]), (\textit{that}, (\mathrm{N}\backslash\mathrm{N})/\mathrm{S}[\mathrm{NP}]), (t, \mathrm{NP}[\mathrm{NP}])\}$$

FIGURE 7 The grammar Eng* used to generate relative clauses.

Here are some examples, with explanations for why the constraints are stated as they are. First, a simple RC in standard tree form.

(51)

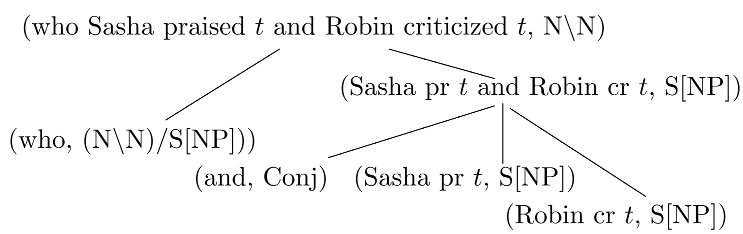


In (51) we see that the trace t has category NP[NP], which we may read as "an NP with an NP gap". It combines with a P_2 on its left to form a P_1 with an NP gap (so the gap feature is passed up). The P_1 with an NP gap combines with an NP on its left to form an S[NP], an S with an NP gap. And now the relative pronoun *who* combines with that to form an N\N. So the gap feature is now eliminated.

Example (51) makes it easy to see why No Vacuous Binding occurs: what follows the wh-word (*who*, *that*, *whose teacher*,...) must, by the category of the wh- words, be of category S[NP]. And the only way to build an S[NP] is to construct an S with a trace t of category NP[NP]; that is, a gap, in some NP position.

To save space in the examples below, we just diagram the RC itself, the part beginning with the wh- word, omitting the initial "every teacher" which is treated the same in all cases. Usually we just give an F–A tree.

(52)



The two conjuncts shown are generated just like the gapped phrase (*Sasha criticized* t, S[NP]) in (51b). Coord applies to the bottom line as the conjuncts have the same, coordinable, category, S[NP]. And we see that the coordinate structure Constraint holds, since if only one conjunct had an NP gap, it would have category S[NP]. But the other would have category S, a different category. So the pair together with (*and*, Conj) would not lie in the domain of Coord. The Across the Board "exception" holds since if all conjuncts have an NP gap, they all have the same category, S[NP]. So the Coord rule applies.

Third, it is also easy to see why we cannot relativize twice into the same clause (Subjacency):

(53) * I see the teacher who$_j$ [John knows the student who$_i$ [t_i criticized t_j]]

The problem is that strings with traces in two argument positions of a given predicate are not derived in our grammar. Consider:

(54) (t criticized t,?)

(t, NP[NP]) (criticized t, (NP\S)[NP])

The strings t and *criticized* t have only the categories indicated, and such a pair does not lie in the domain of any of our FA feature passing rules as they just combine with pairs in which only one element has the feature [NP].

(55) below shows that we can relativize the subject of a sentence complement when there is no immediately preceding complementizer, and (56) indicates that we cannot so relativize when there is a complementizer.

(55) (who Kim said t praised Sasha, N\N)

(who, (N\N)/S[NP]) (Kim said t praised Sasha, S[NP])

(Kim,NP) (said t praised Sasha, NP\S[NP])

(said, (NP\S)/S) (t praised Sasha, S)

(t, NP[NP]) (praised Sasha, NP\S)

Consider what happens when we try to derive a RC of this sort which does have a *that* complementizer. Reading from the top down, we see that the line corresponding to the next to the last one in (55) above would be

(56) (said, (NP\S)/$\overline{\mathrm{S}}$) (that t praised Sasha, $\overline{\mathrm{S}}$[NP])

The only way to derive the right-hand expression would be to combine

(57) (that, $\overline{\text{S}}$/S) (t praised Sasha, S[NP])

But this configuration is precisely the one explicitly ruled out by the condition on the domain of RFA[NP], since (*praised Sasha*, P_1) is an immediate constituent of (*t praised Sasha*, S[NP]). Hence

$$(\textit{that t praised Sasha}, \overline{\text{S}}[\text{NP}])$$

is not derivable in Eng*.

We imposed this condition on the domain of RFA[NP] precisely to block deriving expressions like

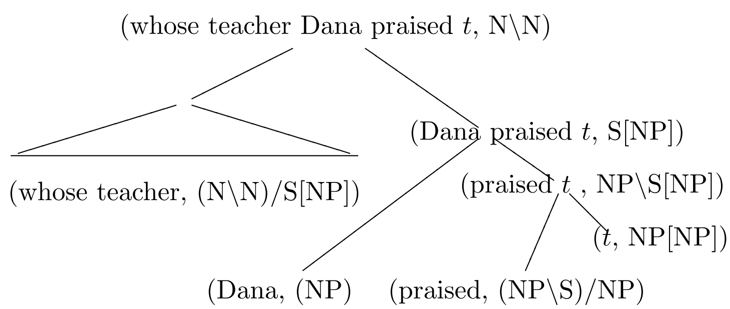
$$({}^{*}\textit{every teacher who Kim said that t praised Sasha}, \text{P}_0/\text{P}_1).$$

This restriction does not follow from some deep principle of grammar. It is rather an ad hoc and fairly superficial condition, one that varies across languages. Spanish, for example, does not really object to relativizing right after a *that*-complementizer:

(58) *Creo que los niños caminaban en el parque.*
I believe that the children were running in the park.

los niños que creo que t caminaban en el parque.
the children that I+believe that *t* were running in the park.

We turn to the matter of relativizing on possessors. Simple Pied Piping works straightforwardly:

(59)


The reason we do not generate expressions such as **the student who John knows t's teacher* is that the trace *t* only has category NP[NP], and *'s* only combines with expressions of category P_0/P_1 or $\text{P}_2\backslash\text{P}_1$. So our grammar does not generate [t's]. (We'll see in Chapter 5 that NP denotations play a role rather different from P_0/P_1 or $P_2\backslash P_1$ denotations).

We are motivated however to extend the lexical entries in (37) as follows:

(60) $(X\backslash(X/\mathrm{N}))$: *'s*, all $X = \mathrm{DP}$, or $((\mathrm{N}\backslash\mathrm{N})/\mathrm{S}[\mathrm{NP}])$

Exercise 4.10. Exhibit an FA tree for

(a student whose teacher's doctor t interviewed her, P_0/P_1).

Lastly, observe that Eng* does generate (61).

(61) (the student whose doctor and whose lawyer Kim interviewed t, N\N)

(Kim interviewed t, S[NP])

(whose doctor and whose lawyer, (N\N)/S[NP])

Coord applies to *whose doctor* and *whose lawyer* in part of (61) which is not shown. The reason is that (N\N)/(S[NP]) is among the coordinable categories, and the expressions are not among those excluded. If either or both are replaced simply by (*who*, (N\N)/S[NP]) or

(*that*, (N\N)/S[NP]),

the resulting triple would not be in the domain of Coord and so cannot be coordinated. We note that this constraint is easy to state on the approach we are proposing, since that approach builds expressions from the bottom up. We start with lexical items and construct increasingly complex expressions by applying our generating functions to them. Of course, we must define the domains of the functions, and to that end we can rule out anything we find motivated.

Exercise 4.11. Exhibit tree derivations (FA or standard) for the expressions whose string coordinates are given below:

a. a teacher who t criticized Sasha
b. every student who Sasha said that Robin thought that Dana praised t
c. a student whose teacher's doctor Sasha interviewed t

Exercise 4.12. Can the grammar of this chapter generate the following strings?

a. Every teacher who Sasha criticized t and who t praised Amy smiles.
b. Every teacher who t likes a student who studies smiles.
c. Every teacher who a student who studies likes t smiles.

If so, give a derivation; if not, say why not.

Exercise 4.13. The ability to assign expressions novel categories has played perhaps an unexpectedly large role in enabling us to formulate original answers to problems of generative syntax. Here is a last curious if not fundamental case. English permits the productive formulation of modifying phrases consisting of an adjective followed by a body part Nwith an -ed suffixed, as in: *a rosy cheeked girl*, *a broad shouldered man*, *a flat footed cop*, etc. Your problem: treating *cheek*, etc. as an N, *rosy*, etc. as an Adj, N/N, design a category for *-ed* such that we generate *rosy cheeked girl*, etc. as an N but do not generate *cheeked girl* as an N.

This completes our illustrative grammar fragment. We turn now in the next chapters to discussion of semantic interpretation.

Further Reading The use of the division notation and the function-argument conception of categories dates from Ajdukievicz (1967). Early foundational works in this style of grammar, called Categorial Grammar, are Lambek (1958) and Bar-Hillel, Gaifman and Bar-Hillel et al. (1960). Wood (1993) is an introductory book on it. A linguistically useful collection of articles on various aspects of categorial grammar is Oehrle, Oehrle et al. (1988). A more recent overview is Chapter 1 in Bernardi (2002). A technically more advanced work is Moortgat (1996).

5

Syntax III: Linguistic Invariants and Language Variation

Work in generative grammar goes well beyond our our simple grammar in Chapter 4. For one, it extends the rules to ones that move, copy and delete constituents. And, our concern here, it attempts to generalize the rules and constraints at any given time to new cases, drawing on diverse languages. Ultimately as linguists we are interested in general properties of human language, not just, say, English with all its peculiarities.

In this chapter (which draws extensively on Keenan and Stabler (2003)) we consider some models of languages in which morphology plays a significantly greater structural role than in English. It enables those languages to ignore the c-command generalization on anaphors mentioned in Chapter 3. To represent this linguistic variation we present a notion of *structural invariant* of a grammar which enables us to generalize across non-isomorphic grammars. It also enables us to show that morphology and lexical items may be "structural" in exactly the same sense in which properties like *is a VP* and relations like *c-commands* are. We draw on notions of invariance used elsewhere in physical science (Chemistry: Cotton (1990); Vision: Mundy and Zisserman (1992). See Weyl (1952) and Gardner (2005) for more general studies. Then we support the following claim:

(1) Anaphora Universals:
For G any grammar of a natural language,
  a. the property of being an anaphor is structurally invariant in G, and
  b. the Anaphor-Antecedent relation is a structural invariant of G.

Of course these claims assume a language independent definition of *anaphor*, the *Anaphor-Antecedent* relation, and *structural invariant*. We turn to the latter task first.

5.1 A Model Grammar and Some Basic Theorems

We present a model grammar *Eng* for English with co-argument anaphora. It is a reduced version of the grammar in Chapter 4, with just enough structural diversity to cover the basic (and a few not so basic) instances of reflexive anaphora in English. It derives sentences like (2) with the constituent bracketing indicated and so satisfies the c-command condition.

(2) a. [John [criticized [himself]]]
    b. [John [criticized [both himself and Bill]]]

Then we present similarly simple models for two languages in which morphology rather than constituency relations controls the distribution of anaphors. We show how to compositionally interpret the morphology permitting anaphors to asymmetrically c-command their antecedents. So the structure of simple sentences in these languages is not isomorphic to those of English.

We exhibit our grammars in a theory-neutral format. We intend that a grammar given in any particular theory—GPSG, LFG, RG, Minimalism, etc—can be presented in this format, so no constraints on expressions derive from the format itself; it is not a theory in the sense in which LFG, Minimalism, etc. are theories.

Definition 5.1. A *grammar* G is a four-tuple $\langle V_G, \mathrm{Cat}_G, \mathrm{Lex}_G, \mathrm{Rule}_G\rangle$, where (omitting subscripts) V and Cat are non-empty sets—the *vocabulary* and *category indices* respectively. The set of *possible expressions* is $V^* \times \mathrm{Cat}$, noted PE_G. Lex, the set of *lexical items* of G, is a finite subset of PE_G, and Rule is a set of *structure building* partial functions of bounded arity from PE_G^* into PE_G. (A function F is of *bounded arity* iff for some n, all $s \in \mathit{Dom}(F)$ are of length $\leq n$.) $\mathcal{L}(G)$, the *language generated by* G, is the closure of Lex_G under the $F \in \mathrm{Rule}_G$.

5.1.1 The grammar Eng

V: laughed, cried, sneezed, praised, criticized, punished, congratulated, John, Bill, Sam, Ed, himself, and, or, nor, both, either, neither

Cat: P_0, P_1, P_2, $\mathrm{P}_{01}/\mathrm{P}_{12}$, $\mathrm{P}_1/\mathrm{P}_2$, CJ

Lex: $\mathbf{P_1}$**:** laughed, cried, sneezed (I.e. (laughed, P_1) $\in \mathrm{Lex}_{\mathrm{Eng}}$, etc.)
$\mathbf{P_2}$**:** praised, criticized, punished, congratulated
$\mathbf{P_{01}/P_{12}}$**:** John, Bill, Tom, Ed ($\mathrm{P}_{01}/\mathrm{P}_{12}$ is often noted NP)
$\mathbf{P_1/P_2}$**:** himself

CJ: and, or, nor

Rule: {Merge, Coord}, defined below for s and t arbitrary elements of V^*.

Merge:

$$(s, A), (t, B) \to (s \frown t, \mathrm{P}_0) \quad A = \mathrm{P}_{01}/\mathrm{P}_{12}, B = \mathrm{P}_1,$$
$$(s, A), (t, B) \to (t \frown s, \mathrm{P}_1) \quad A \in \{\mathrm{P}_1/\mathrm{P}_2, \mathrm{P}_{01}/\mathrm{P}_{12}\}, B = \mathrm{P}_2.$$

We understand from this notation (here and later) that Merge is a two place function. Its domain is the set of pairs of possible expressions mentioned on the left above. Its value at each argument is given at the head of the arrow. The Function-Argument tree in (3) summarizes the argument that (John laughed, P_0) is in $\mathcal{L}$(Eng).

(3) Merge: (John laughed, P_0)

- (John, $\mathrm{P}_{01}/\mathrm{P}_{12}$)
- (laughed, P_1)

The leaf nodes are lexical items and since $\mathcal{L}$(Eng) is closed under Merge, which applies to the pair of leaf nodes, we infer that (John laughed, P_0) $\in \mathcal{L}$(Eng). The tree is just a pictorial representation of the argument that (John laughed, P_0) $\in \mathcal{L}$(Eng). It has no status in our definition of grammar. Our second rule is Coordination:

Coord:

$$(\text{and}, CJ)(s, C)(t, C) \to (\text{both } s \text{ and } t, C) \quad C \in \text{Cat} - \{\text{CJ}\},$$
$$(\text{and}, CJ)(s, C)(t, C') \to (\text{both } s \text{ and } t, \mathrm{P}_1/\mathrm{P}_2) \quad C \neq C', \text{ and } C, C' \in \{\mathrm{P}_1/\mathrm{P}_2, \mathrm{P}_{01}/\mathrm{P}_{12}\}.$$

(4) Merge: (Ed criticized both himself and Bill, P_0)

- (Ed, $\mathrm{P}_{01}/\mathrm{P}_{12}$)
- Merge: (criticized both himself and Bill, P_1)
    - (criticized, P_2)
    - Coord: (both himself and Bill, $\mathrm{P}_1/\mathrm{P}_2$)
        - (and, CJ)
        - (himself, $\mathrm{P}_1/\mathrm{P}_2$)
        - (Bill, $\mathrm{P}_{01}/\mathrm{P}_{12}$)

Exercise 5.1.

a. Write out the rules introducing *either…or…* and *neither…nor…*.

b. Exhibit function-argument derivation trees for:
   i. John both laughed and criticized either himself or Bill.
   ii. Ed neither criticized himself nor punished both himself and Bill.
   iii. Neither John nor Bill criticized both himself and Ed.

Note that *John* and *himself* differ in category. $(\text{himself cried}, \mathrm{P}_0) \notin \mathcal{L}(\text{Eng})$, nor is $(\text{both himself and John cried}, \mathrm{P}_0)$.

5.1.2 Some general syntactic notes

1. Different grammars may use different categories. In Eng they are ad hoc but mnemonic: P_n is n-place predicate. Expressions of category P_0, zero place predicates (they require 0 arguments to make a sentence) are interpreted as True or False.

2. For an expression $s = (w, C) \in \mathrm{PE}_G, \mathrm{Cat}(s) =_{df} s_2$, the second coordinate of s, and $\mathrm{string}(s) =_{df} s_1$. This notation facilitates defining the $F \in$ Rule and renders trivial the identification of the category of an expression. Also it avoids much lexical ambiguity; e.g. $(\text{respect}, \mathrm{N})$ and $(\text{respect}, \mathrm{V})$ are distinct lexical items—same string coordinate but different category coordinates. Analogously for *honor*, *judge*, *desire*, *envy*, *love*, etc. Also, some rules, such as Type Lifting: $(\mathrm{Ed}, \mathrm{NP}) \to (\mathrm{Ed}, \mathrm{S}/(\mathrm{NP}\backslash\mathrm{S}))$ and rules deriving deverbal nouns and denominal verbs: $(\textit{shoulder}, \mathrm{N}) \to (\text{shoulder}, \mathrm{V})$ just target the category coordinate. Others may primarily affect the string coordinate, such as Reduplication (Malagasy), Preposition + Article fusion (German: *in* + *das* = *ins* 'in the'; Hebrew: *bə* + *ha* = *ba* 'in the', Italian: *de* + *il* = *del* 'of-the', Greek *se* + *to* = *sto* 'to-the', etc.) and like-form constraints (Spanish: *Les* 'them.dat' → *se* when immediately followed by a *l*-initial pronoun: **Les la di* → *Se la di* 'to.them it I-gave'.

3. We require a compositional semantics, so derived expressions are interpreted as a function of the interpretations of what they are derived from and the functions used to derive them. For most lexical items we must learn their meanings de novo, and only finitely many independent meanings can be so learned, so Lex is finite.

4. *The complexity hierarchy* is the chain

$$\mathrm{Lex}_0 \subseteq \mathrm{Lex}_1 \subseteq \cdots$$

given by:

$$\mathrm{Lex}_0 = \mathrm{Lex}_G, \text{ and for all } n,$$
$$\mathrm{Lex}_{n+1} = \mathrm{Lex}_n \cup \{F(s) \mid F \in \mathrm{Rule}_G \ \&\ s \in \mathrm{Lex}_n^* \cap \mathit{Dom}(F)\}.$$

So Lex_1 is $\mathrm{Lex}_0 (= \mathrm{Lex}_G)$ plus all expressions obtained by applying Merge and Coord to appropriate sequences of expressions in Lex_0. Lex_2

is Lex_1 plus all expressions obtained by applying Merge and Coord to appropriate sequences of expressions from Lex_1, etc.

Theorem 5.1.

a. $\mathcal{L}(G) = \bigcup_{n\in\mathbb{N}} Lex_n$, *and*
b. $Rule_G$ *is finite* $\rightarrow$ *each* Lex_n *is finite.*

We turn now to the crucial notion of an *automorphism* of a grammar. Informally, an automorphism of G is a way of substituting expressions for expressions without changing how expressions are derived. The substitution must map distinct expressions to distinct expressions, and each expression in $\mathcal{L}(G)$ must have something mapped to it.

Definition 5.2. An *automorphism* of a grammar G is a bijection $h : \mathcal{L}(G) \rightarrow \mathcal{L}(G)$ which fixes each $F \in \text{Rule}_G$, that is, $h(F) = F$. Aut_G is the set of automorphisms of G.

So treating $F \in \text{Rule}_G$ as a set of sequences, $F = h(F)$ just means that $F = \{h(d) \mid d \in F\}$. To say $F(a_1, \ldots, a_n) = b$ is just to say that $\langle a_1, \ldots, a_n, b\rangle \in F$, so $\langle h(a_1, \ldots, a_n), h(b)\rangle \in h(F) = F$. Thus, $F(h(a_1, \ldots, a_n)) = h(b) = h(F(a_1, \ldots, a_n))$, so h commutes with F, $h \circ F = F \circ h$.

Exercise 5.2. For F and h as above, show that $h(F) = F$ if h commutes with F.

Fact. We define an $h \in \text{Aut}_G$ just by giving its values on lexical items. Its values on all derived expressions follow, given that $h(F(s_1, \ldots, s_n)) = F(h(s_1), \ldots, h(s_n))$.

Notation. For $f : A \rightarrow B$, f extends to a map (also noted f when no confusion results) from A^* into B^* by setting $f(\langle a_1, \ldots, a_n\rangle) = \langle f(a+1), \ldots, f(a_n)\rangle$. And if f has been extended to a set D then f extends to the power set of D by setting $f(X) = \{f(x) \mid x \in X\}$.

$\text{Perm}(A)$ is the set of permutations of A—the bijections from A to A, and $|A|$ is the cardinality of A Since Lex_G is finite and $|\text{Aut}_G| \leq$ $mboxPerm(\text{Lex}_G)$, it follows that

Theorem 5.2. Aut_G *is finite.*

Exercise 5.3. For A finite with $|A| = n$, $|\text{Perm}(A)| = n!$. Say why.

Definition 5.3. A *group* $\mathbf{A}$ is a four-tuple $\langle A, \cdot, ^{-1}, e\rangle$, where A is a set, the *domain* of $\mathbf{A}$, $\cdot$ is a binary function on A, $^{-1}$ is a unary function on A, and e is an element of A, which satisfy the three conditions below for all $x, y, z \in A$:

a. Associativity: $((x \cdot y) \cdot z) = (x \cdot (y \cdot z))$,
b. Identity: $(x \cdot e) = (e \cdot x) = x$, and

c. Inverses: $((x^{-1} \cdot x) = (x \cdot x{-}1) = e$.

Theorem 5.3. *Aut_G contains the identity map id_G on $\mathcal{L}(G)$ and is closed under function composition and inverses and is thus a* group.

So whenever $h \in \mathrm{Aut}_G$ then so is h^{-1} (which maps y to x iff h maps x to y), and whenever $g, h \in \mathrm{Aut}_G$ so is $h \circ g$, which maps x to $h(g(x))$. Aut_G is called the *automorphism group of* G and also the *symmetry group of* G.

Here are some more examples of groups. The first is infinite, but our interest lies in finite groups since Aut_G is finite for any grammar G.

(5) The set $\mathbb{Z}$ of integers (positive, negative and 0) with $\cdot = +$, $e = 0$, and $n^{-1} = -n$. To verify this claim we check that $+$ is associative: $((n + m) + p) = (n + (m + p))$; 0 is an identity element: $n + 0 = 0 + n = n$; and $-$ is the inverse $^{-1}$: $n + -n = -n + n = 0$.

(6) The set $\{1, -1\}$ with $\times$ (multiplication) as the binary operation. What is the identity element? For each $n \in \{1, -1\}$, what is n^{-1}?

(7) Consider a regular pentagon P (regular = all angles the same, all sides the same length) with vertices named a, b, c, d, e in counterclockwise order. A *rotation* of P rotates P around the geometric center moving each vertex a given multiple of 72 deg so that each vertex moves to the spot where another vertex was. There are clearly just 5 rotations, $r_0, r_1, \ldots, r_4$ where each r_i maps each node i vertices ahead. We may represent the five rotations in the table below.

$$\begin{array}{rccccc} x = & a & b & c & d & e \\ r_0(x) = & a & b & c & d & e \\ r_1(x) = & b & c & d & e & a \\ r_2(x) = & c & d & e & a & b \\ r_3(x) = & d & e & a & b & c \\ r_4(x) = & e & a & b & c & d \end{array}$$

So each r_i is a a permutation of the vertices (but there are $5! = 120$ permutations of the vertices, so most permutations are not rotations). In fact the set of the r_i is a group, where $r_j \cdot r_i$ informally is first apply r_i then apply r_j.

Exercise 5.4. For the rotation group in the above example,

a. i. Define the $\cdot$ operation explicitly.
ii. Which r_i is the identity element?

iii. For each r_i, what is its inverse?

b. For each vertex v let f_v be the permutation of P obtained by reflecting the pentagon around the line bisecting the angle subtended by v. The bisecting line meets the opposite side perpendicularly. For example, f_a is given by:

$$\begin{array}{rccccc} x = & a & b & c & d & e \\ f_a(x) = & a & e & d & c & b \end{array}$$

i. Complete the table, giving values for $f_b(x), f_c(x), f_d(x)$, and $f_e(x)$.

ii. The set of these reflections does not form a group under function composition. Give a sufficient reason why not.

iii. If we add the five reflections to the set of five rotations we do get a group (called a dihedral group). Exhibit by table each of the following: $r_1 \circ f_a$, $f_a \circ r_1$, $r_2 \circ f_b$, and $f_a \circ f_a$.

Exercise 5.5. Let $\mathbb{Q}^+$ be the set of positive fractions, and let $\cdot = \times$ (multiplication). What choice can we make for e and the inverse function so that the result is a group?.

Exercise 5.6. Let A be a non-empty set and $\text{Perm}(A)$ the set of permutations of A. We claim that $\text{Perm}(A)$ is (the domain of) a group, where $\cdot = \circ$ (function composition), $e = \text{id}_A$, the map sending each $a \in A$ to itself, and $^{-1}$ is function inverse (that is, by definition h^{-1} maps a to b iff h itself maps b to a).

a. State the three things you must prove to show that $\text{Perm}(A)$ above is a group.

b. Prove each of those three statements.

Theorem 5.4 (Cayley)**.** *Every group is isomorphic to a group of permutations of a set A. (Though this group may just be a proper subset of Perm(A)).*

Exercise 5.7. Let A be an arbitrary group. Prove each of the following (always understood as universally quantified). We often do half the problem, which serves as a hint about how to do the other half. We sometimes write xy for $x \cdot y$.

a. Right cancellation: $xa = ya \to x = y$,
Proof: Assume $xa = ya$. Then

$$\begin{array}{ll} (xa) \cdot a^{-1} = (ya) \cdot a^{-1} & (\cdot \text{ is a function}) \\ x(a \cdot a^{-1}) = y(a \cdot a^{-1}) & (\cdot \text{ is associative}) \\ xe = ye & (\text{axiom on inverses}) \\ x = y & (\text{axiom on } e\ ) \end{array}$$

b. State and prove the left cancellation law.
c. Prove that $e^{-1} = e$.
d. Uniqueness of e. Let z satisfy condition 2 in Definition 5.3, that is, for all x, $x \cdot z = z \cdot x = x$. We show that $z = e$. Since $xz = x$, all x, then $xz = xe$, so by left cancellation, $z = e$. Your problem: Suppose that $zx = x$, all x. Show that $z = e$.
e. Uniqueness of inverses. We show more: namely, for each $x \in A$ there is a unique y such that $x \cdot y = e$. Let x be given. Suppose that $x \cdot y = e$. Then $x \cdot y = x \cdot x^{-1}$, so by left cancellation $y = x^{-1}$, which is what we wanted to show. Your problem: Show that for each $x \in A$ there is a unique y such that $y \cdot x = e$.
f. Prove that $(x^{-1})^{-1} = x$.
g. Prove that $(x \cdot y)^{-1} = y^{-1} \cdot x^{-1}$.

Definition 5.4. We define a binary relation $\simeq$, *is structurally equivalent to*, on $\mathcal{L}(G)$ by:

$$s \simeq t \text{ iff there is an } h \in \mathrm{Aut}_G \text{ such that } h(s) = t.$$

So expressions d and d' in some $\mathcal{L}(G)$ have the same structure iff there is an $h \in \mathrm{Aut}_G$ which maps d to d' (whence h^{-1} maps d' to d). Since by definition automorphisms do not change the structure building functions this is a pretheoretically reasonable way to characterize sameness of structure. It also decides some cases which our pretheoretical intuitions leave undecided: Compare the sentences in (8).

(8) a. John criticized Bill.
b. John criticized himself.

In Eng these expressions are derived in exactly the same way, differing just by a lexical item. If we replace *himself* in (8a) by *Ed* for example there is an automorphism that derives one from the other. But no automorphism can map one of (8) to the other. The reason (omitting several steps) is because no automorphism can map an NP, say $(\mathrm{Bill}, \mathrm{P}_{01}/\mathrm{P}_{12})$, to $(himself, \mathrm{P}_1/\mathrm{P}_2 1)$ since such an automorphism or a variant of it would provably map $(\mathrm{Bill\ laughed}, \mathrm{P}_0)$ to $(\mathrm{himself\ laughed}, \mathrm{P}_0)$, which is not in $\mathcal{L}(\mathrm{Eng})$, contradicting that the range of an automorphism of Eng is $\mathcal{L}(\mathrm{Eng})$. We note the following theorem.

Theorem 5.5. *The relation $\simeq$ is an equivalence relation. Writing $[s]_G$ for the equivalence class of s, we have $[s]_G =_{df} \{t \in \mathcal{L}(G) \mid s \simeq t\}$.*

So $[s]_G$ is the set of expressions with the same structure as s. Equivalence relations are studied more explicitly in the next chapter. Here we just note that to say that $\simeq$ is an equivalence relation means: (1) for all $s \in \mathcal{L}(G)$, $s \simeq s$; (2) for all $s, t \in \mathcal{L}(G)$, $s \simeq t \rightarrow t \simeq s$, and (3) for all $s, t, u \in \mathcal{L}(G)$, $((s \simeq t \;\&\; t \simeq u) \rightarrow s \simeq u)$. These are all natural properties for a "sameness of structure" relation to have. Note that we have defined sameness of structure between expressions without saying what "the" structure of any expression is.

We turn now to the crucial notion of an invariant. Loosely, they are properties of the grammar which are unchanged under any way of substitution of expressions for expressions which preserves how expressions are derived. Formally,

Definition 5.5. A linguistic object d over G is *structurally invariant* iff for all $h \in \mathrm{Aut}_G$, $h(d) = d$. (We generalize this definition slightly later).

Definition 5.6. A *linguistic object* over G is an element of $\mathcal{L}(G)$, a subset of $\mathcal{L}(G)$ (that is, a property of expressions), a relation on $\mathcal{L}(G)$, a (partial) function from $\mathcal{L}(G)^*$ to $\mathcal{L}(G)$, etc.

Definition 5.7. A *fixed point* of a function F is an object b in its domain such that $F(b) = b$.

Definition 5.8. The *invariants* of a grammar are the *fixed points* of its syntactic *automorphisms*.

Theorem 5.6. *The following are invariants of Eng:*

a. *Lex_{Eng}, in fact for each n, Lex_n,*
b. *All $PH(C)$, where $PH(C) =_{df} \{s \in \mathcal{L}(Eng) \mid Cat(s) = C\}$, and*
c. *$(himself, P_1/P_2)$ but not $(john, P_{01}/P_{12})$.*

Theorem 5.7. *The $PH(C)$ and $(himself, P_1/P_2)$ are invariant.*

The proof begins with three lemmas.

Lemma 5.8. *Automorphisms fix $\subseteq$ (and all operations definable in terms of it), that is $X \subseteq Y \rightarrow h[X] \subseteq h[Y]$.*

Proof. If $b \in h[X]$ then $b = h(x)$, for some $x \in X$, so $b = h(y)$, for some $y \in Y$ (namely $y = x$), so $b \in h[Y]$. Since b was arbitrary, $h[X] \subseteq h[Y]$. □

Lemma 5.9. *For $A \subseteq \mathcal{L}(G)$, if $h(A) \subseteq A$, for all automorphisms h, then $h(A) = A$, for all automorphisms h.*

Proof. Assume the antecedent and let h be arbitrary in Aut. Then $h^{-1}(A) \subseteq A$. Since h preserves subset (by lemma 5.8) $h(h^{-1})(A)) \subseteq h(A)$. But $h(h^{-1}(A)) = A$, so $A \subseteq h(A)$, completing the proof (since h was arbitrary). □

Lemma 5.10. *Lex_{Eng} is invariant.*

Proof. Suppose for some $s \in \text{Lex}, h(s) \notin \text{Lex}$. Then $h(s)$ is the value of Merge or Coord at some arguments, hence $h^{-1}(h(s)) = s$ is also the value of Merge or Coord at some arguments, hence its string coordinate is a concatenation of two or more vocabulary items. But that is false, no element of Lex_{Eng} has such a string coordinate. Hence $h(s) \in Lex$, proving the lemma. □

We now proceed to the proof of theorem 5.7.

Proof. We do PH(NP) below, and the other categories in the Addendum to this chapter. Inspection of Merge shows that:

a. If $s \in \text{Range(Merge)}$ then $\text{Cat}(s) = \text{P}_0$ or $\text{Cat}(s) = \text{P}_1$.
b. If $\text{Cat}(s) = \text{NP}$ then $\text{Cat}(h(s)) = \text{NP}$, for all $h \in \text{Aut}_{\text{Eng}}$. Choose t and u of category NP and P_2 respectively. Then $\langle s, \text{Merge}(t, u)\rangle \in$ *Dom*(Merge). So $\langle h(s), h(\text{Merge}(t, u))\rangle = \langle h(s), \text{Merge}(h(t), h(u)\rangle \in$ *Dom*(Merge). Since no expression of category P_0 is a coordinate in *Dom*(Merge), $\text{Merge}(h(t), h(u))$ has category P_1, so the only possibility for $\text{Cat}(h(s))$ is NP, as was to be shown. Thus PH(NP) is invariant.
c. $h(\text{himself}, \text{P}_1/\text{P}_2) = (\text{himself}, \text{P}_1/\text{P}_2)$. Note that by lemma 5.10, $h(\text{himself}, \text{P}_1/\text{P}_2) \in \text{Lex}$, and by lemma 7 (Addendum), it has category P_1/P_2. The only candidate is $(\text{himself}, \text{P}_1/\text{P}_2)$.

□

So $(\text{himself}, \text{P}_1/\text{P}_2)$ is a grammatical morpheme in Eng. No structure preserving map can replace it by anything else. Note that (and, CJ) is not fixed by all automorphisms; it can be mapped to (or, CJ) and (nor, CJ), but those are the only possibilities.

5.2 A Semantic Definition of Anaphor

Studying Eng we see that $(\text{himself}, \text{P}_1/\text{P}_2)$ shares many properties with NPs. They all combine with P_2s to form P_1s. And they coordinate with each other. Still, *himself* (we omit the category coordinate when unnecessary) in distinction to NPs, does not combine with P_1s to form P_0s. Now we intend that *himself* is an anaphor. But suppose we were Martians just discovering English—how would we know that it was an anaphor, and not simply an NP with a slightly restricted distribution—though the restriction extends to boolean compounds which contain it: (*neither himself nor John* also does not combine with P_1s to form P_0s). What we need is a way of identifying expressions as anaphors independently of their category name or syntactic distribution on pain of making claims about their distribution circular. We now provide such a way.

5.2.1 Generalized Quantifiers

Given a domain E, a *generalized quantifier* F over E maps $\mathcal{P}(E)$, the power set of E, into $\{\text{False}, \text{True}\}$, usually noted $\{0, 1\}$. F extends *accusatively* to maps F_{acc} from $\mathcal{P}(E \times E)$ to $\mathcal{P}(E)$ by $F_{\text{acc}}(R) = \{a \in E \mid F(aR) = 1\}$, where $aR =_{df} \{b \in E \mid (a, b) \in R\}$. For example,

(9) For A and B subsets of E, EVERY(A) maps B to 1 (True) iff $A \subseteq B$. So *Every poet daydreams* is True iff POET $\subseteq$ DAYDREAM, that is, the set of poets is a subset of the set of daydreamers.

Similarly SOME(A) maps B to 1 iff $A \cap B \neq \emptyset$; NO$(A)(B) = 1$ iff $A \cap B = \emptyset$ and MOST$(A)(B) = 1$ iff $|A \cap B| > |A|/2$ (A assumed finite).

In general R-expressions, which we call *Referentially Autonomous* expressions, denote generalized quantifiers, and their value at a binary relation (denoted say by a transitive verb phrase) is determined by the values they assign to the subsets of E, as given in above. The maps F_{acc} below are just those that satisfy the Accusative Extensions Condition (AEC), Keenan (1987b).

The Accusative Extensions Condition (AEC): For all $a, b \in E$, all $R, S \subseteq E \times E$, if $aR = bS$ then $a \in F(R)$ iff $b \in F(S)$.

The denotation of the higher order *most of John's students* in (10a) satisfies the AEC:

(10) a. Sam criticized most of John's students.
b. Sam criticized himself.

If the people Sam criticized are just those who Bob praised then *Sam criticized most of John's students* has the same truth value as *Bob praised most of John's students*.

But the denotation of *himself* fails the AEC: if Sam criticized just Fred, Mark, Bob and Ben, and those are just the people Bob praised, then *Sam criticized himself* is False and *Bob praised himself* is True. But the denotation of *himself* does satisfy the weaker Accusative Anaphor Condition (AAC) below:

The Accusative Anaphor Condition (AAC): For all $a \in E$, all $R, S \subseteq E \times E$, if $aR = aS$ then $a \in F(R)$ iff $a \in F(S)$

An appropriate denotation for *himself* in $\mathcal{L}$(Eng), as well as the grammars we provide for Korean and Toba Batak shortly, is SELF:

(11) SELF$(R) = \{a \in E \mid (a, a) \in R\}$

One verifies that for any E with $|E| \geq 2$, SELF satisfies the AAC and fails the AEC. So will the denotation of conjunctions and disjunctions of *himself* with proper nouns in all those languages. This yields a language independent semantic definition of anaphor.

Definition 5.9. An element $u \in \mathcal{L}(G)$ is an *anaphor* iff all its non-trivial[1] denotations satisfy AAC and fail AEC.

Fact. The set of anaphors in $\mathcal{L}(\mathrm{Eng})$ is exactly $\mathrm{PH}(\mathrm{P}_1/\mathrm{P}_2)$, so the property of being an anaphor in Eng is invariant.

Further expressions containing anaphors are interpreted compositionally as in Chapter 4. We note that even in our utterly simple grammar antecedents may occur arbitrarily far from their anaphors:

(12) John either praised Ed or both laughed and criticized both himself and Bill.

Definition 5.10. In $\mathcal{L}(\mathrm{Eng})$, s is a *possible antecedent* of an anaphor t in u (or $s\mathrm{AA}t$ in u iff $\mathrm{Cat}(s) = \mathrm{P}_{01}/\mathrm{P}_{12}$ and there is a constituent v of category P_1 in u such that s is sister of v and t is a proper constituent of v.

In (12) both *himself* and *both himself and Bill* are anaphors, and *John* is the antecedent of each of them according to our definition. It is immediate from the definition that antecedents of anaphors in $\mathcal{L}(\mathrm{Eng})$ c-command them. That condition is not sufficient however, as *Bill* is not a possible antecedent of *himself* in *both Bill and himself.*

Our definition of the AA relation above is ultimately unsatisfactory, though it is significant that the cases it identifies we judge pretheoretically correct. Still it doesn't apply to grammars that lack a category called $\mathrm{P}_{01}/\mathrm{P}_{12}$. So it wouldn't apply to the trivial variant of Eng in which '$\mathrm{P}_{01}/\mathrm{P}_{12}$' is everywhere replaced with 'DP'. We don't (quite) have a language independent definition analogous to that for anaphor. It is thus nearly trivial that the AA relation above is invariant since it is defined as a logical compound of invariant properties and relations. (We show below that constituency relations and their boolean compounds are universally invariant).

Open Problem 5.1. *Find a language independent definition of the AA relation.*

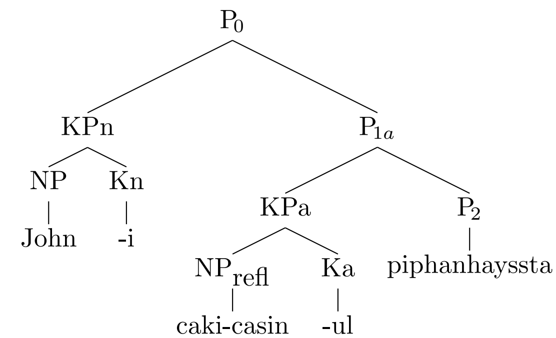
5.3 A Model of Korean

We turn now to a model Kor of Korean illustrating a different, but still structurally invariant way of presenting anaphors.

[1] F is *trivial* iff either for all relations R, $F(R) = \emptyset$ or for all R, $F(E) = E$.

Kor models case marking and word order in Korean: Verbs are final in their clauses and NPs are suffixed with (here) one of two case markers, *-nom* and *-acc*. The resulting phrases we call Kase Phrases (KPs). KPs are freely ordered preverbally with no topicality difference. Kor illustrates how morphology can be directly structural in lieu of c-command. The anaphor asymmetrically c-commands its antecedent in (13b).

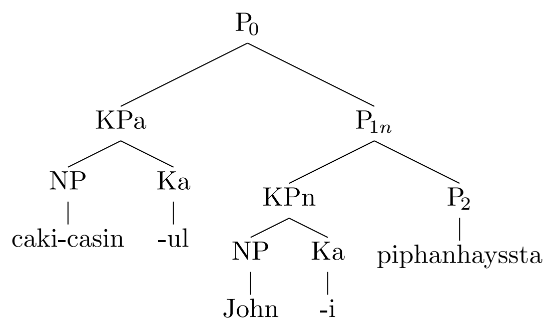
(13) a.



John	*-i*	*cacki-casin*	*-ul*	*piphanhayssta*
John	-nom	self	-acc	criticized

John criticized himself

b.



cacki-casin	*-ul*	*John*	*-i*	*piphanhayssta*
self	-acc	John	-nom	criticized

John criticized himself

To get a better sense of the basic role of morphology here let us see first that the *acc*-first order in (13b) differs dramatically from English object-first orders as in *Himself John likes (but no one else)*. For one, in English object-first orders with reflexives are not very natural with non-individual denoting arguments such as quantified or interrogative ones. But in both (13a) and (13b) *John* can be replaced preserving naturalness with *nwuka* 'who?' or *motun haksayng-tul* 'all the students'.

(14) a. *Caki-casin-ul* *nwuka* / *motun* *haksayng-tul-i*
self-acc who / all student-pl-nom
piphanhayssta
criticized

Who criticized himself? / All the students criticized themselves

b. ??Himself who criticized?

c. ??Himself every student criticized.

Secondly, the object-first order in English is largely a root clause phenomenon. We cannot for example relativize the subject after fronting the object:

(15) a. Himself the man likes

b. *the man who himself likes

But relativizing from a reflexive first order in Korean is natural:

(16) a. *Caki-casin-ul* *John-i* *hoyuy-eyse* *piphanhayssta*
Self-acc John-nom meeting-loc criticized

John criticized himself at the meeting

b. *[Caki-casin-ul* *John-i* *piphanhay-n* *hoyuy-ka]*
Self-acc John-nom criticize-sub meeting-nom
ecey *iss-ess-ta*
yesterday be-pst-decl

There was a meeting yesterday at which John criticized himself

(Nom marking in Korean is *-i* with consonant final Nouns, *-ka* with vowel final ones. Acc marking *-ul* vs *-lul* is similarly conditioned, as is topic marking *-un* vs *-nun*).

Third, topicalization effects in Korean are achieved by morphological means with a topic marker *-(n)un* rather than syntactic fronting:

(17) *John-i* *Ed-un* *piphanhayssta*
John-nom Ed-topic criticized

John criticized ED

Finally, while the relative preverbal order of case marked arguments is free, what we can't naturally change is the morphological marking:

(18) **Motun* *haksayng-tul-ul* *caki-casin* *-i* *piphanhayssta*
All student-pl-acc self -nom criticized

All the students criticized themselves

Also the result of replacing *motun haksayng-tul-ul* 'all the students -acc' with other accusatively marked arguments such as *kwukwu-lul* 'who-acc?' or *john-ul* 'John-acc' remains ungrammatical.

The moral of these observations is that naively Korean syntax uses bound morphology, case and topic marking, in a way not present in English. So the observable syntax of simple clauses in these two languages is systematically different.

Here is a grammar of minimal main clauses with reflexives, comparable to Eng, which illustrates case marking invariants and direct compositional interpretation of Ss with anaphors c-commanding their antecedents. Note that we have "dissimilated" the P_1 categories so that once a KP in a certain case combines with a P_2 then no KP in that same case can combine with the resulting P_1.

5.3.1 The grammar Kor

Cat: NP, NP_{refl}, Ka, Kn, KPa, KPn, P_2, P_{1a}, P_{1n}, P_0, CJ

Lex: **NP:** John, Bill, Sam, Kim

NP_{refl}: self

Kn: -nom

Ka: -acc

CJ: and

P_{1n}: laughed, cried sneezed

P_2: praised, criticized, teased

CJ: and

Rule: CM (Case Mark), PA (Predicate-Argument), Coord

CM:

Domain		Value	Conditions
(-nom,Kn)(t, NP)	$\rightarrow$	$(t\text{-nom}, \text{KPn})$	none
(-acc,Ka)(t, C)	$\rightarrow$	$(t\text{-acc}, \text{KPa})$	$C \in \{\text{NP}, \text{NP}_{\text{refl}}\}$

PA:

Domain		Value	Conditions
$(s, \text{KP}x)(t, \text{P}_{1n})$	$\rightarrow$	$(s \frown t, \text{P}_0)$	$x \in \{n, a\}$
$(s, \text{KP}x)(t, \text{P}_2)$	$\rightarrow$	$(s \frown t, \text{P}_1)$	$x \neq y \in \{n, a\}$

Coord:

Domain		Value	Conditions
(and, CJ)t,Cu,C	$\rightarrow$	$(t \frown \text{ and } \frown u, C)$	$C \in \{\text{NP}, \text{NP}_{\text{refl}}, \text{P}_0, \text{P}_{1n}, \text{P}_{1a}, \text{P}_2\}$
(and, CJ)t,Cu,C'	$\rightarrow$	$(t \frown \text{ and } \frown u, \text{NP}_{\text{refl}})$	$C \neq C' \in \{\text{NP}, \text{NP}_{\text{refl}}\}$

So (self-acc john-nom praised, P_0) and (john-nom self-acc praised, P_0) $\in \mathcal{L}$(Kor).

5.3.2 Some invariants of Kor

a. (-nom, Kn), (-acc, Ka) and (self, $\mathrm{NP}_{\mathrm{refl}}$) but not (bill, NP). So the case marking suffixes are structural in Korean, as linguists naively assume. Thus our formal account captures our basic intuitions here.

b. For all $C \in \mathrm{Cat}, \mathrm{PH}(C)$.

c. For each n, Lex_n.

d. Anaphors (as defined semantically earlier). Provably the anaphors in Kor are just the expressions s with $\mathrm{Cat}(s) = \mathrm{KPa}$ and (self, $\mathrm{NP}_{\mathrm{refl}}$) as a constituent.

e. The AA relation, where: for s, t constituents of u, $s\mathrm{AA}t$ in u iff t is an anaphor, $\mathrm{Cat}(s) = \mathrm{KPn}$ and for some constituent v of u, either

    i. v has category P_{1n}, s is a sister of v and t is a constituent of v, or

    ii. v has category KPa, t is a constituent of v and for some w of category P_{1a}, v is a sister of w and s is a constituent of w.

We omit proofs for reasons of space, but we emphasize that whether a linguistic object of a grammar is invariant is a matter of proof, not pretheoretical intuition. One does not ponder long and hard to conclude "Hmmm. I can get (-nom, Kn) as invariant".

Notice too that, like (4) in English, (19) contains two anaphors and *John-nom* antecedes both.

(19) (John-nom [[[Bill and self]-acc] [criticized]])

On the other hand, $\mathcal{L}(\mathrm{Kor})$ presents an anaphoric possibility not present in $\mathcal{L}(\mathrm{Eng})$: what we are calling a P_{1a} (an accusative taking P_1) in (13b) can coordinate:

(20) *Caki-casin* *-ul* *[John-i* *piphanha-ko* *Bill-nom*
Self -acc John-nom criticize-and Bill-nom
chingchanhayssta]
praised

John criticized and Bill praised himself

Here both *John-i* and *Bill-i* are antecedents of *caki-casin-ul.* An analogous type of binding in English would be Right Node Raising sentences, as in (21b) (if grammatical).

(21) a. [John bought but Bill cooked] [the turkey]
b. ?[John punished but Bill congratulated][himself]

Note lastly an essential similarity between the distribution of anaphors in $\mathcal{L}(\mathrm{Eng})$ and $\mathcal{L}(\mathrm{Kor})$. In Eng anaphors can combine with two place predicates to make one place ones, but can not combine with one place ones to form Sentences (truth bearing expressions). In $\mathcal{L}$(Kor) the analogous restriction is given by case marking: Anaphors can be case marked accusative, but not nominative[2] in main clauses. Hence they cannot combine to form main clauses with intransitive predicates. In this sense then Korean handles morphologically a constraint that English handles syntactically.

5.3.3 Interpreting anaphors in $\mathcal{L}$(Kor)

We interpret (self, $\mathrm{NP}_{\mathrm{refl}}$) as SELF, as with (himself, $\mathrm{P}_1/\mathrm{P}_2$) in $\mathcal{L}$(Eng). We interpret (*-acc*, Ka) as an identity function, so (John-*acc*, KPa) denotes the same individual that (John,NP) denotes. A KPn maps $\mathcal{P}(E)$ to truth values as expected, but maps binary relations R to maps taking type-1 functions into $\{0,1\}$ by: $F(R)(G) = F(G(R))$. So (KIM-nom(R))(SELF) = (KIM-nom)(SELF(R)), the correct interpretation. (*-nom*, Kn) maps NP denotations to KPn denotations. So the entire complication here rests with the interpretation of a grammatical formative, (*-nom*, Kn).

5.4 Toba Batak

Toba Batak (see Schachter (1984), Cole and Hermon (2008), Keenan (2009)) is a simple example of a structure type characteristic of West Austronesian languages (Tagalog in the Philippines, Malagasy in Madagascar). The languages are verb initial and voice marking, "dual" to case marking. Voice affixes (primarily *maN-* and *di-* in Toba) combine with verbal roots determining the structure of the clause. Anaphors may asymmetrically c-command their antecedents (23b).

(22) a. *[mang-ida si Ria] si Torus*
AF-see art Ria art Torus
Torus sees Ria

b. *[di-ida si Torus] si Ria*
PF-see art Torus art Ria
Torus saw Ria

(23) a. *[mang-ida dirina] si Torus*
AF-see self art Torus
Torus sees himself

a'. *mang-ida si Torus dirina

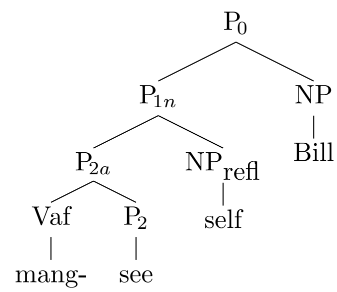
b. *[di-ida si Torus] dirina*
PF-see art Torus self

[2]This holds for main clauses, but as is well known, bare *caki* can occur nominatively in a complement clause, bound to the subject of the matrix verb.

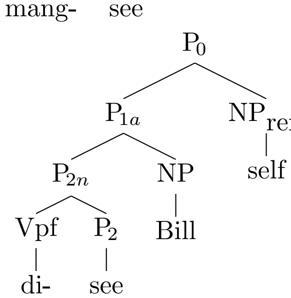
Torus saw himself

b'. *di-ida dirina si Torus

(24) a.



b.



The papers in Schachter (1984) provide very strong support for the major constituent break in (24a-b). We note that both *mang-ida* and *di-ida* require two arguments. And with both verbs the immediate postverbal argument cannot be separated from the verb by adverbs like *nantoari* 'yesterday'. More dramatically, the immediate postverbal NP cannot be relativized or questioned by movement, only the clause final NP above can. Consider the paradigm below (which holds mutatis mutandis in Tagalog and Malagasy).

(25) a. *Manjaha* *buku* *guru* *i*
read book teacher Det
'The teacher reads the book'

b. *Dijaha* *guru* *buku* *i*
read teacher book Det
'The teacher read the book'

c. *guru* *na* *manjaha* *buku* *i*
teacher lnk read book Det
'the teacher who is reading a book'

d. *buku* *na* *dijaha* *guru* *i*
book lnk read teacher Det
'the book the teacher read'

e. **buku* *na* *manjaha* *guru* *i*
   book lnk read teacher Det

f. **guru* *na* *dijaha* *buku* *i*
   teacher lnk read book Det

Judgments on (25e-f) are strong and immediate. They reflect the basic interpretative mechanisms of simple clauses, not just arbitrary constraints on what can be extracted. (25f) for example can only mean the teacher that a book read, which is nonsense. Finally Emmorey (1984) provides spectographic evidence for the major constituent break in (24a-b). She shows that the verb, whether a *mang-* or a *di-* one, plus its following DP form an intonation group in which the nuclear pitch accent falls on the stressed syllable of the last lexical item in the Predicate Phrase (PredP). Pitch accent placement identifies the immediate post-verbal DP in *mang-* and *di-* verbs as the last lexical item in their PredP's. In conjoined PredP's both conjuncts receive this accent (Emmorey op.cit.):

(26) a. *[[Manuhor* *baoang]* *jala* *[mangolompa* *mangga]]* *halak* *an*
   Buy onions and cook mangos man Det

   The man buys onions and cooks mangos

b. *[[Dituhor* *si* *Ore]* *jala* *[dilompa* *si* *Ruli]]* *mangga*
   di-buy art Ore and di-cook art Ruli mangos

   Ore buys and Ruli cooks mangos

5.4.1 A Grammar for Toba Batak

Cat: Vaf, Vpf, P_2, P_{2a}, P_{2n}, P_{1a}, P_{1n}, P_0, NP, NP_{refl}, CJ

Lex: **Vaf:** mang-
Vpf: di-
P_2: praised, criticized, saw
P_{1n}: laughed, cried
NP: john, bill
NP_{refl}: self
CJ: and, or

Rule: VM (Verb Mark), PA (Predicate-Argument), Coord

VM:

Domain		Value	Conditions
(mang, Vaf)(t, P_2)	$\rightarrow$	(mang$\frown t, P_{2a}$)	none
(di, Vpf)(t, P_2)	$\rightarrow$	(di$\frown t, P_{2n}$)	none

PA:

Domain		Value	Conditions
$(s, \mathrm{P}_{2x})(t, \mathrm{NP})$	$\to$	$(s \frown t, \mathrm{P}_{1y})$	$x \neq y \in \{n, a\}$
$(s, \mathrm{P}_{1x})(t, \mathrm{NP})$	$\to$	$(s \frown t, \mathrm{P}_0)$	$x \in \{n, a\}$
$(s, \mathrm{P}_{2a})(t, \mathrm{NP}_{\mathrm{refl}})$	$\to$	$(s \frown t, \mathrm{P}_{1n})$	none
$(s, \mathrm{P}_{1a})(t, \mathrm{NP}_{\mathrm{refl}})$	$\to$	$(s \frown t, \mathrm{P}_0)$	none

Coord:

Domain		Value	Conditions
$(\text{and}, \mathrm{CJ})(t, C)(u, C)$	$\to$	$(t \frown \text{ and } \frown u, C)$	$C \in \{\mathrm{NP}, \mathrm{NP}_{\mathrm{refl}}, \mathrm{P}_0, \mathrm{P}_{1n}, \mathrm{P}_{1a}, \mathrm{P}_2\}$
$(\text{and}, \mathrm{CJ})(t, C)(u, C')$	$\to$	$(t \frown \text{ and } \frown u, \mathrm{NP}_{\mathrm{refl}})$	$C \neq C' \in \{\mathrm{NP}, \mathrm{NP}_{\mathrm{refl}}\}$

5.4.2 Some Invariants of Toba Batak

We write x *con* y, here and later, for x *is a constituent of* y.

a. (*mang*-, Vaf), (*di*-, Vpf) and (*self*, $\mathrm{NP}_{\mathrm{refl}}$). So as with Korean, the bound morphemes are structural invariants, coinciding again with our intuitions as linguists.
b. Lex; in fact all Lex_n.
c. For $C \in \mathrm{Cat}$, $\mathrm{PH}(C)$.
d. The set of anaphors in Toba. (We note without proof that the set of anaphors in Toba is just $\mathrm{PH}(\mathrm{NP}_{\mathrm{refl}})$).
e. $s\mathrm{AA}t$ in u iff $\mathrm{Cat}(s) = \mathrm{NP}$ and either there is a constituent v of u with $\mathrm{Cat}(v) = \mathrm{P}_{1n}$, s sister v and t con v or there is a t' con v with $\mathrm{Cat}(v) = \mathrm{P}_{1a}$ and t con t' and s con v.

5.4.3 Interpreting anaphors in Toba Batak

Briefly: (*self*, $\mathrm{NP}_{\mathrm{refl}}$) denotes SELF, as expected, the rest is determined by the denotations of the voice affixes: $\text{MANG}(R)(G)(F) = F(G(R))$ and $\text{DI}(R)(G)(F) = G(F(R))$. So e.g. $\text{DI}(\text{SEE})(\text{JOHN})(\text{SELF}) = \text{JOHN}(\text{SELF}(\text{SEE}))$. So as in Kor, interpreting Ss with asymmetrically c-commanding anaphors is unproblematic.

5.5 Some Mathematical Properties of Grammars and their Invariants

There are at least two reasons for presenting grammars in a mathematically explicit way. One, it enables us to formulate clearly pretheoretical notions that motivate our work as linguists. And two, we can study mathematically explicit models of grammars, proving theorems about them, thereby extending our knowledge of language structure. Often a precise mathematical formulation of an issue allows us to ask new questions, ones we could not have really formulated without the mathematical notions and notation at hand. In the first category we suggest the following three ideas.

5.5.1 Degree of grammaticization

The *degree of grammaticization* of a linguistic object is the percent of automorphisms that fix it. Since Aut_G is finite it makes sense to speak of percentage here. So it makes sense to say that Prepositions and Conjunctions in English are more grammaticized than common nouns but less grammaticized than personal pronouns.

5.5.2 Historical grammaticization

Commonly, grammatical morphemes derive historically from content words, such as common nouns and verbs. In several languages such as Basque, Hausa, Georgian and Tamazight Berber the content word *head* came to be interpreted as self (see Heine and Kuteva (2002)). We might (partially) characterize that evolution as a progressive decrease in the percentage of automorphisms that "move" the item (that is, map it to something else). At the end point of change no automorphisms would move it, that is, all would fix it.[3] So in Berber "his head" sometimes means himself and sometimes his head. For further reading, see Givon (1971), Hopper and Traugott (1993), Bybee et al. (1994).

5.5.3 Grammatical morphemes

We have proposed a rigorous characterization of grammatical morphemes as ones fixed by all the automorphisms of the grammar. Here we generalize that idea, prompted by an observation in Keenan and Stabler (2003) concerning masculine and feminine noun classes in Spanish.[4] Adjectives and determiners take agreement markers which vary with the noun class of noun they combine with and it turns out that for some natural choices of lexicon some automorphisms systematically interchange masculine and feminine nouns (also interchanging the agreement markers). But we still want to say that the property of being a masculine (feminine) noun is a structural property even though not all automorphisms fix it. So we shall distinguish between variable automorphisms and non-variable ones, which we call stable.

Definition 5.11. For f and g functions,

a. g is a *restriction* of f iff $Dom(g) \subseteq Dom(f)$ and for all $x \in$

[3] A more precise statement would be more complicated. As is well acknowledged (Hopper (1991), Heine and Reh (1984)) a grammaticized item may also retain its original meaning.

[4] We are indebted to Philippe Schlenker (pc) for pushing us on this point.

$Dom(g), g(x) = f(x)$.

b. f *extends* g iff g is a restriction of f.

c. If $B \subseteq Dom(f)$, $f|B$ is that restriction of f with domain B.

Definition 5.12. For $G = \langle V_G, \mathrm{Cat}_G, \mathrm{Lex}_G, \mathrm{Rule}_G \rangle$,

a. For each $S \subseteq V_G \times \mathrm{Cat}_G$, $G[S] =_{df} \langle V_G, \mathrm{Cat}_G, \mathrm{Lex}_G \cup S, \mathrm{Rule}_G \rangle$. When S is a singleton $\{s\}$ we write simply $G[s]$ for $G[\{s\}]$.

b. For $s \in V_G \times \mathrm{Cat}_G$, G is *free* for s iff $s \notin \mathcal{L}(G)$ and there is an $h \in \mathrm{Aut}_{G[s]}$ such that for some $t \in \mathrm{Lex}_G$, h interchanges s and t and fixes all other $u \in \mathrm{Lex}_{G[s]}$. The intuition: G being free for s means we can add s to G without changing any of the pre-existing structural relations between the elements of $\mathcal{L}(G)$[5].

c. G is *free* for $S \subseteq V_G \times \mathrm{Cat}_G$ iff for all $s \in S$, G is free for s and $G[s]$ is free for $S - \{s\}$. (Note that all G are free for $\emptyset$).

d. $h \in \mathrm{Aut}_G$ is *stable* iff for all finite S for which G is free there is a $k \in \mathrm{Aut}_{G[S]}$ such that k extends h.

e. A linguistic object d over G is a *stable invariant* iff $h(d) = d$, for all stable $h \in \mathrm{Aut}_G$.

We note that the automorphisms of $\mathcal{L}(\mathrm{Span})$ which can interchange the masculine and feminine Nouns and the agreement markers are not free for (poet,$\mathrm{N}_{\mathrm{fem}}$). Observe also that the stable invariants of G are always a superset (not necessarily proper) of the invariants of G since they are only required to be fixed under a subset of the automorphisms of G, namely the subset of stable automorphisms.

Thesis 5.1. *The grammatical morphemes of a grammar are the lexical items which are stable invariants.*

Theorem 5.11. *The set of stable automorphisms of a grammar G contain the identity function, are closed under inverses and composition, and thus form a subgroup of Aut_G.*

Our second motivation for presenting grammars mathematically is that by proving theorems about them we gain knowledge of their formal structure. Theorem 5.11 above is already one example. We exhibit some others below.

5.6 Invariants of Type 0

The *type 0 invariants* of a grammar G are the expressions fixed by the stable automorphisms. Are there any *universal* invariants of type 0? To find some we might look for uniformly defined semantic objects,

[5] Adding new lexical items typically results in new derived expressions. Recall that the structure building functions are defined on $V^* \times \mathrm{Cat}$.

like the function SELF, and see if we feel that an expression in any natural language which denotes one of those objects is a stable invariant. Negation is a candidate. In the next chapter we define the notion *boolean complement*, often (but not exclusively) denoted in English by *not* or *n't*. We call an expression a *negation* if it always denotes boolean complement. Then we claim:

(27) Claim: For all natural languages G, if $d \in L(G)$ is a negation then d is a stable invariant.

Note that (27) does not claim that all natural languages have an expression denoting negation. This is largely true, but, it seems, not quite always. Old Tamil can express negation just by eliminating the tense morpheme on the verb (see Pederson (1993)).

Equally several languages (French, Hausa, Middle English) may use discontinuous expressions of negation, whose interpretation would have to be studied carefully—are the parts independently meaningful? Or is it just one expression which is very superficially split into two phonological parts? Note too that (27) just claims that if d is a negation in a language then it is syntactically distinguished, but it doesn't say that its syntactically distinctive properties are the same in different languages, just as the coding of anaphors may be distinctive in different languages (some may code it in terms of case marking, others in terms of voice marking, etc).

In a way similar to negation we feel that other boolean operators are conditional invariants. Namely, boolean greatest lower bound operators (see Chapter 8) denoted in English by *and* and *every/all*, and least upper bound operators, denoted by *or* and *some*.

(28) Claim: For all natural language G, if $d \in L(G)$ always denotes a boolean greatest lower bound (least upper bound) operator then d is a stable invariant.

There seem to us a variety of other semantically defined operations which are commonly expressed as type 0 invariants (though we have studied none of these in detail as yet): monomorphemic expressions of case, voice, tense, number, gender, cause, verbal nominalization (e.g.gerundive *-ing*). This claim assumes, non-trivially, language independent definitions of these notions and then of course detailed investigation of the grammars of many languages.

Finally, the candidates for type 0 invariants proposed above are all *empirical invariants* in the sense that none of them (not even the boolean ones) follow from our definition of invariant. But below we see there are some invariants of types 1 and 2 which are universal in this sense, that is, they hold of all G regardless of whether G is a candidate for modeling a natural language or not. We call these *uniform invariants* as opposed to empirical invariants.

5.7 Invariants of Type (1)

Invariants of type (1) are subsets of $L(G)$, that is, properties of expressions.

5.7.1 Universal uniform invariants ($\mathbf{INV}_1$)

These follow from our definitions of grammar, automorphism, and invariant.

Theorem 5.12. *For all grammars G:*

a. $\emptyset$ *and* $L(G)$ *are invariant.*

b. For all $F \in Rule_G$, $Dom(F)$ *and* $Ran(F)$ *are invariant. Intuitively much "structure" is imposed by what the structure building functions can apply to.*

c. $[s]_G$ *is invariant, for all* $s \in L(G)$*. No non-empty proper subset of* $[s]_G$ *is invariant.*

d. If no $s \in Lex_G$ *is derived* ($Lex_G \cap Ran(F) = \emptyset$ *for all* $F \in Rule_G$) *then each* $Lex_n \in INV_1$*.*

e. (Closure Conditions) INV_1 *is closed under relative complement and arbitrary unions and intersection. So it is a complete atomic Boolean algebra, (See Chapter 7). For example, if the properties of being a feminine Noun and of being a plural Noun are invariant then so are the properties of being a feminine plural Noun and a feminine non-plural Noun.*

Exercise 5.8. Describe the atoms of INV_1 for arbitrary G. (Assumes some knowledge of Boolean Algebra. See Chapter 8.

5.7.2 Universal empirical invariants

These are ones that hold for models G for a natural language NL but may fail for artificially constructed G that satisfy our definitions.

(29) Claim: The set of anaphors of NL is a stable invariant.

In general it is interesting to determine which semantically defined sets of expressions are "coded in syntax", i.e. invariant.

(30) The set K of nouns denoting aquatic mammals in English is not invariant. We find no grammatical difference between *porpoise* and *shark*, so there is an automorphism h of English which switches them, so $h(K) \neq K$.

(31) Is the set of polar (Yes/No) questions in English invariant? We don't know. A Yes answer would mean that whenever φ is a polar question so is $h(\varphi)$, for all automorphisms h of English. And do we expect that in any NL the property of being a polar question is invariant?

(32) Is the set of monotone decreasing DPs in English invariant? Very possibly. Such DPs (Keenan (1996), Chapter 10 here) include *no doctor*, *neither John nor Mary*, *at most ten students*, *fewer than ten students*, *no student's doctor*, *not more than ten boys*, *neither applicant*.

5.7.3 On the role of categories

For $C \in \mathrm{Cat}_G$, $\mathrm{PH}(C)$, the set of phrases of category C, may fail to be invariant, per our discussion of gender classes of Nouns in Spanish (but also Latin, Kinyarwanda).

Theorem 5.13 below however does yield one condition, which arises in the study of formal languages and models of artificial languages (Propositional Logic, First Order Logic), which does imply that each $\mathrm{PH}(C)$ is invariant.

Definition 5.13. G is *category functional* iff for each $F \in \mathrm{Rule}_G$, each $u \in L(G)^* \cap Dom(F)$, $\mathrm{Cat}(F(u))$ is a function of the categories of $u_1, \ldots, u_{|u|}$.

So the category of a derived expression is predictable from the function F used to derive it and the categories of the arguments of F.

Theorem 5.13. *Given G, if each $Lex(C)$, the lexical expressions of category C, is invariant, and G is category functional then all $PH(C)$ are invariant.*

Otherwise, we the most we can claim is that sameness of category is invariant:

Definition 5.14. G is *category uniform* iff for all $s, t \in \mathcal{L}(G)$, if $\mathrm{Cat}(s) = \mathrm{Cat}(t)$ then for all $h \in \mathrm{Aut}_G, \mathrm{Cat}(h(s)) = \mathrm{Cat}(h(t))$.

Category Uniformity does not follow from our definition of invariant, but it is a condition on the role of categories that seems to us satisfied in many cases. We tentatively propose it as an axiom that NLgrammars must satisfy.

Exercise 5.7.3.1. *Show that Category Uniformity can be strengthened to an "if and only if".*

When G is category uniform each $h \in \mathrm{Aut}_G$ lifts to a permutation h' of $\{\mathrm{PH}(C) \mid C \in \mathrm{Cat}\}$, where $h'(\mathrm{PH}(C)) = \mathrm{PH}(C')$ iff for some

$s \in \mathrm{PH}(C), h(s) \in PH(C')$.

Automorphisms which uniformly permute sets of expressions represent natural *linguistic symmetries*. And an important syntactic role of grammatical categories is to provide the structural means of expressing these regularities. That is, if an automorphism h maps one member of some $\mathrm{PH}(C)$ to a member of some $\mathrm{PH}(C')$, then h must map all $s \in \mathrm{PH}(C)$ to $\mathrm{PH}(C')$. Gender classes are a common linguistic symmetry. And we observe (Corbett (1991)) that languages with gender classes normally exhibit gender agreement of various kinds: Adj+Noun, Det+Noun, Subject+Predicate, etc. "Why, after all, would a language have many noun classes and not use them for anything?" Latin and Russian have three gender classes, and more noun classes when the imperfect cross product with declension classes is taken into account. Kinyarwanda (Kimenyi (1980)), typical for Central Bantu, has 16 noun classes, several the plurals of others.

Natural languages present many symmetries in addition to noun classes. **Conjugation** classes in Romance come to mind. Thus along with many irregular verbs, Spanish has three regular verb classes distinguished according as their infinitives end in *-ar*, *-er*, *-ir*. Structurally deeper however are *voice* classes—active vs passive. In W. Austronesian (Tagalog, Malagasy) there may be a half dozen voice distinctions, not just a twofold active/passive one. *Mood* classes—indicative vs subjunctive, and *aktionsart* classes—active vs stative, accomplishment vs achievement, are further candidates.

The linguistic symmetries of G induce additional structure on Aut_G according as they permute whole $\mathrm{PH}(C)$ or just expressions within each $\mathrm{PH}(C)$.

5.8 Invariants of Type (2) and higher

5.8.1 Universal uniform invariants ($\mathbf{INV}_2$)

The $F \in \mathrm{Rule}_G$ are trivially invariant as automorphisms by definition fix them.

Moreover, several of the syntactic conditions investigated in generative grammar over the last two generations ("island constraints", such as Coordinate Structure, Subjacency, etc.) can be stated as invariant properties of the $Fin\mathrm{Rule}_G$. For example

a. If $F \in \mathrm{Rule}_G$ is a movement rule then F satisfies island constraints.

b. If $F \in \mathrm{Rule}_G$ is a copy rule (Kandybowicz (2008), Kobele (2006)) F does not iterate. (For example, usually reduplication rules do not iterate; but see Blust (2001)).

Note that these last two conditions concern the action of the structure building functions—obviously a critical domain to investigate, but not one we have discussed until now.

INV_2, the set of *invariant binary relations*, includes $\emptyset$ and $L(G) \times L(G)$, and like INV_1 is a complete atomic boolean algebra—in fact it is a proper *relation algebra* as it is closed under composition, converse, and contains the identity relation.

If $A, B \in \text{INV}_1$ then $A \times B$ and $[A \to B]$, the set of functions:$A \to B$, are in INV_2.

Structural equivalence, $\simeq$, and logical equivalence, $\equiv$, are independent, where we say that sentences φ and ψ are *logically equivalent* iff they have the same truth value in all situations (models), a notion we explore in more detail in the next chapter.

Proof. In one direction consider that s and t below are logically equivalent.

$$\begin{aligned} s &= (\text{Exactly half of American males are overweight}, \text{S}) \\ t &= (\text{Exactly half of American males are not overweight}, \text{S}) \end{aligned}$$

But t is not structurally equivalent to s; its predicate is the negation of that of s.

Going the other way, consider s and t below:

$$\begin{aligned} s &= (\text{Seven boys sang eight songs each}, \text{S}) \\ t &= (\text{Eight boys sang seven songs each}, \text{S}) \end{aligned}$$

As there seem to be no grammatical differences between *seven* and *eight* it is reasonable that an automorphism of English could interchange these sentences, so they are structurally equivalent. But obviously they are not logically equivalent. □

But there are other uniformly definable relations of major linguistic interest that are not prominent in the more general algebraic setting. Fundamental here are constituency relations. We define *immediate constituent* (ICON), *proper constituent* (PCON), and *constituent* (CON) below:

Definition 5.15.

a. $s\text{ICON}t$ iff for some $F \in \text{Rule}_G$, some $u \in L(G)^* \cap Dom(F)$, $t = F(u)$ and $s = u_i$, for some $1 \leq i \leq |u|$.
b. $s\text{PCON}t$ iff for some $n > 1$, there is a $u \in L(G)^n$ with $s = u_1$, $t = u_n$ and each $u_i\text{ICON}u_{i+1}$ for $1 \leq i < n$.
c. $s\text{CON}t$ iff $s = t$ or $s\text{PCON}t$.

d. These relations are invariant in all G, as are ones defined in terms of them, such as *sister of* and *c-command*: s *sister* t in u iff for some constituent v of u, sICONv, tICONv and $s \neq t$. s *c-commands* t in u iff t is a constituent of a sister of s in u.

5.8.2 Remarks on constituency relations

Firstly, we do not define them in terms of trees. The use of standard trees in generative grammar only depicts derivations for a very restricted class of concatenative functions. They cannot represent substitution functions, such as Montague's quantifying which derived *Every student likes some teacher* from ((*some teacher*), (*every student likes* v)) by substituting *some teacher* for v (see Montague (1974)) . We do expect however that our constituency analysis would coincide with usual ones in a grammar whose rules just concatenate arguments, inserting fixed lexical items if needed.

Second, we can now see how our grammars—algebras of partial functions—differ massively from ordinary numerical algebras. Consider for example the set $\mathbb{Z}$ of integers (positive, negative and zero) under the binary function $+$. Since any two integers can be added, every pair of integers (n, m) is in the domain of $+$. That is, $+$ is a total function. Now, let n be an arbitrary integer. We claim that any integer m is a "constituent" of n. That is, there is an integer x such that $+(m, x) = n$. Just choose $x = (n - m)$. But a derived expression in a natural language never has all expressions in the language as constituents.

Third, CON as defined is reflexive and transitive but may fail to be antisymmetric, contrary to the ordinary usage of *is a constituent of*. That is, it may be that sCONt and tCONs but that $s \neq t$. To see this in the general algebraic setting consider the squaring function ^{2} (on the positive real numbers) and the positive square root function $\sqrt{\ }$. 3CON9 since $3^2 = 9$, and 9CON3 since $\sqrt{9} = 3$. But $9 \neq 3$. As an artificial linguistic case let $f, g \in$ Rule with $g(x, A) = (xb, A)$ and $f(yb, A) = (y, A)$, all $x, y \in V^*$. So g suffixes a b and f erases string final b's. So (x, A)CON(xb, A) by g, and (xb, A)CON(x, A) by f, but $(x, A) \neq (xb, A)$. Note that in this case, (x, A)PCON(x, A).

In practice we find no convincing cases where an expression, especially a lexical item, is a proper constituent of itself. Pullum (1976) studies a properly broader class of cases he calls *Duke of York Derivations*. His examples are mostly phonological and critiqued in McCarthy (2003), who points out that the B forms in the apparent A-B-A derivations are not well formed as is, so we lack motivation for deriving them. Pullum notes one syntactic case in which an operation in Italian inserts a clitic that is later deleted after triggering verb agreement. But the expression derived by the deletion isn't identical to the one into which the clitic was inserted since it has undergone verb agreement and no longer hosts clitic insertion. The sequence of operations which apparently leads to an sPCONs configuration does not iterate yielding derivations $s \to \cdots \to s \to \cdots$, so we lack a clear case of sPCONs. We propose then *Foundation* as an axiom of natural languages grammars:

Axiom 5.1. For all natural language G:

a. **Full Foundation**: for all $s, t \in L(G)$, sPCON$t \to s \neq t$.
b. **Lexical Foundation**: for all $s \in \mathrm{Lex}_G$, all $t \in L(G)$, sPCON$t \to s \neq t$.

Lexical Foundation is weaker than Full Foundation (and properly entailed by it). But it suffices for our later purposes, as automorphisms are decided by their behavior on the lexical items. We note further:

Theorem 5.14.

a. G satisfies Full Foundation $\to$ CON is antisymmetric.
b. Lexical Foundation $\to$ Eliminability of derived lexical items.

Proof.

a. Lemma: PCON is transitive. Let sPCONt and tPCONu. Then the derivation of t from s followed by that of u from t is a derivation of u from s in $n > 1$ steps, so sPCONu.
Now, let sCONt & tCONs. Assume for contradiction that $s \neq t$. Then sPCONt and tPCONs, so by the lemma, sPCONs. By Full Foundation $s \neq s$, a contradiction. So $s = t$.
b. Let G satisfy Lexical Foundation with $s \in \mathrm{Lex}_G$ derived, so $s = F(u)$, for some $u \in L(G)^*$. Let $G\backslash s$ be that grammar like G except $\mathrm{Lex}_{G\backslash s} = \mathrm{Lex}_G - \{s\}$. Then $L(G\backslash s) = L(G)$ and $\mathrm{Aut}_G = \mathrm{Aut}_{G\backslash s}$. Suppose for some i, sCONu_i. Then sPCONs since $s = F(u)$, so by Lexical Foundation $s \neq s$, a contradiction. So $\neg s$CONu_i any i. So any derivation of $F(u)$ in $L(G)$ consists entirely of expressions in $L(G\backslash s)$, so $s \in L(G\backslash s)$ whence $L(G\backslash s) = L(G)$ and trivially $\mathrm{Aut}_G = \mathrm{Aut}_{G\backslash s}$.

□

Corollary 5.15. *Since Lex_G is finite, induction using Theorem 5.14b allows us to remove all derived lexical items without changing $L(G)$ or Aut_G. So a G with Lexical Foundation behaves like a G with no derived lexical items (whence Lex_G and thus all Lex_n are invariant).*

Lexical Foundation also implies a deeper property, *Bounded Structure* (Keenan and Stabler (2003)). The idea is that the structural complexity of a natural language is determined by a finite initial fragment. Beyond a certain complexity level more complex expressions iterate structure already known. We can state this intuition explicitly. We begin by treating each Lex_n as a language in its own right:

Definition 5.16. For each n, a *local automorphism* of Lex_n is a map $h : \mathrm{Lex}_n \to \mathrm{Lex}_n$ satisfying:

a. h is bijective,
b. for each $F \in \mathrm{Rule}_G$, $h fixes Dom(F_n)$, where $F_n =_{df} (F|\mathrm{Lex}_n)$, and
c. if $n > 0$ then $h(F_n(t)) = F_n(h(t))$, all $t \in Lex^*_{n-1} \cap Dom(F)$

For $h \in \mathrm{Aut}_G$, $h|\mathrm{Lex}_n$ is a local automorphism of Lex_n. But there may be other local automorphisms of Lex_n that do not extend to h' in Aut_G. In our model of Korean Lex_0 has NPs, case markers, and P_1. NPs and case markers combine to form KPn's and KPa's, nominative (and accusative) Kase Phrases, in Lex_1. Then in Lex_2 KPns but not KPas combine with P_1 to form P_0. So local automorphisms of Lex_0 can interchange the *-nom* and *-acc* case markers but they don't extend to automorphisms in general, not even to automorphisms of Lex_2. And Bounded Structure says that for some n, the automorphisms of Lex_n are just the restrictions of the $h \in \mathrm{Aut}_G$ to Lex_n. So in this sense the structures of expressions in $L(G)$ is expressed by a finite subset of $L(G)$.

Theorem 5.16 (Bounded Structure)**.** *If G satisfies Lexical Foundation then there is an n such that the automorphisms of Lex_n are exactly the automorphisms of G restricted to Lex_n.*

5.8.3 The role of categories in recursion

A last approach to the types of recursion in natural language is built on the notion of a *derivational cycle*. Let us say that a *cycle* is a sequence $C \in \mathrm{Cat}_n$, for some $n > 1$ with $C_1 = C_n$, such that for every positive integer k there is a $t \in L(G)$ which embeds $k{\cdot}C = (C_1, \ldots, C_{n-1})^{k-1}, C$. That is, there is a sequence of $n+(n-1)(k-1)$ expressions with each s_i of category C_i an immediate constituent of s_{i+1}, and the last expression in the sequence is an immediate constituent of t.

For examplei, a grammar of English might have $C = \langle \mathrm{NP}, \mathrm{Det}, \mathrm{NP} \rangle$ as a cycle. Then an expression of English which embeds $1 \cdot C$ would be *John's teacher*, where *John* has category NP, *John's* has category Det, and *John's teacher* has category NP. (33a) below embeds $2 \cdot C$, and (33b) embeds $3 \cdot C$.

(33) a. John's teacher's doctor
b. John's teacher's doctor's father

Axiom 5.2. Cyclicity: All G for natural languages have a cycle.

Cyclicity guarantees that $L(G)$ is infinite as long as the last expression instantiating a cycle $(C_m, \ldots, C_{m+n-1})$ is longer than the first. Full Foundation has the same effect.

5.8.4 Universal empirical invariants?

Plausible cases, assuming language independent definitions of the relevant notions, are:

a. Category Uniformity (sameness of grammatical category)
b. The Anaphor-Antecedent relation
c. Agreement relations (gender, number, case, definiteness)[6]

Lastly, we also hypothesize that *Theta Role assignment* is invariant. Theta roles (Agent, Recipient, Patient,...) are relations between individuals and the properties and relations expressed by predicates. Somewhat more precisely:

(34) For s, t constituents of u: $\theta(s, t)$ iff sPCONt and s is assigned θ in t but not in any proper constituent of t.

(35) a. John praised the teacher
Agent(John, John praised the teacher)
Theme (the teacher, praised the teacher)
b. The teacher was criticized by John
Agent(John, was criticized by John)
Theme (the teacher, the teacher was criticized by John)
c. The teacher's criticism of John
Agent(the teacher, the teacher's criticism of John)
Theme (John, criticism of John)

We claim that the θ relations are invariant: $\theta(s, t)$ iff $\theta(hs, ht)$, for all automorphisms h. So if expressions are isomorphic their corresponding constituent pairs stand in the same theta relations. So if the theta roles of corresponding constituent pairs in (36a,a') are different then (36a,a') are not isomorphic. Similarly for (36b,b'):

(36) a. John produced the play
a'. John enjoyed the play
b. John danced
b'. John vanished

[6]See Corbett (1991) and Barlow and Ferguson (1988) for some of the possibilities.

In contrast, theta equivalence does not imply structural equivalence. Each of (35a,b,c) is *theta equivalent* to the others in the sense of presenting the same number of pairs with the same theta roles but none is isomorphic to any of the others. And of course we would like a more rigorous definition of theta role assignment.

5.9 Structure Preserving Operations on Grammars

Such operations are ones which do not change $L(G)$ or Aut_G but they may change derivations. The most important is *generalized composition.*

Definition 5.17. For $F, H \in \mathrm{Rule}_G$, $F \circ_i H$ is defined by:

$$F(s_1, \ldots . s_{i-1}, H(t_1, \ldots, t_m), s_{i+1}, \ldots, s_n),$$

for each

$$(t_1, \ldots, t_m) \in \mathit{Dom}(H) \text{ and each}$$
$$(s_1, \ldots s_{i-1}, H(t_1, \ldots, t_m), s_{i+1}, \ldots, s_n) \in \mathit{Dom}(F).$$

Here $F \circ_i H$ is the ith *composition* of F with H.

Theorem 5.17. *Closing* Rule_G *under* generalized composition *preserves structure.*

Compare now the two derivations for (Sam praised Ed,P_0) in Eng closed under ith compositions:

(37) a. Merge: (Sam praised Ed, P_0)

(Sam, NP) Merge: (praised Ed, P_1)

(Ed, NP) (praised, P_2)

b. Merge $\circ_2$ Merge: (Sam praised Ed, P_0)

(Sam, NP) (Ed, NP) (praised, P_2)

Each non-terminal node in the left-hand tree is binary branching, the only non-terminal node in the right hand tree is ternary branching. Adding compositions of rules to Rule_G does not lose derivations, it just adds new ones. Given G, write G° for the grammar that results from closing Rule_G under generalized composition (ith compositions, all i).

Corollary 5.18. *If* $L(G)$ *is infinite and* Rule_G *finite (our typical case) then*

a. Every derived $s \in L(G^\circ)$ *has a derivation tree of depth 1 (and* *usually many others of greater depth). Every non-derived $s \in L(G^\circ)$ has a derivation tree of depth 0.*

b. For every n there is an $m \geq n$ such that some expression of $L(G)$ has a derivation tree of depth 1 whose root node is m-ary branching.

c. $Rule_{G^\circ}$ is infinite.

Corollary 5.18 suggests rethinking the significance of the binary branching requirement (Kayne (1984)) on derivations. If G is binary branching and Rule_G is finite then just adding in (not closing) some $\{F \circ_i H \mid F, H \in \mathrm{Rule}_G\}$ results in a G still with just finitely many rules, but often some derivations with ternary branching nodes, as with (Sam praised Ed,P_0) above. So given a binary branching derivational system we can construct an extensionally (same expressions) and structurally (same automorphisms) equivalent one which fails to be binary branching.

But Corollary 5.18 is just one datum influencing the form of rules, learnability is another, more important one. Plus, too many cases in nature seem to opt for just one among two or more symmetric (automorphic) alternatives: Life forms are built only from right spiraling DNA, its left spiraling counterpart can be created in the lab and is stable, and so does not violate any basic law of nature. Why are there not two types of life forms—right spiraling and left spiraling ones? The physicist Feynman can wonder "Why is nature so nearly symmetric?" (Bunch (1989)[pg. 189]).

5.10 Addendum

We complete the proof that the $\mathrm{PH}(C)$ for $C \in \mathrm{Cat}_{\mathrm{Eng}}$ are invariant.

3. Show that $\mathrm{Cat}(h(u)) = \mathrm{P}_2$ if $\mathrm{Cat}(u) = \mathrm{P}_2$. Let s, t both of category NP. Then $\langle s, \mathrm{Merge}(t, u)\rangle \in \mathit{Dom}(\mathrm{Merge})$ and for $h \in \mathrm{Aut}_{Eng}, h(s, \mathrm{Merge}(t, u)) = \langle hs, \mathrm{Merge}(ht, hu)\rangle \in \mathit{Dom}(\mathrm{Merge})$, so $\mathrm{Cat}(\mathrm{Merge}(ht, hu) = \mathrm{P}_1$, so $\mathrm{Cat}(h(u)) = \mathrm{P}_2$. Thus $\mathrm{PH}(\mathrm{P}_2)$ is invariant.
4. $\langle(\mathrm{and}, \mathrm{CJ}), (\mathrm{praised}, \mathrm{P}_2), (\mathrm{criticized}, \mathrm{P}_2)\rangle \in \mathit{Dom}(\mathrm{Coord})$ so for $h \in \mathrm{Aut}_{\mathrm{Eng}}$ arbitrary $\langle h(\mathrm{and}, \mathrm{CJ}), h(\mathrm{praised}, \mathrm{P}_2), h(\mathrm{criticized}, \mathrm{P}_2)\rangle \in \mathit{Dom}(\mathrm{Coord})$ since $h(\mathrm{PH}(\mathrm{P}_2)) = \mathrm{PH}(\mathrm{P}_2)$ then $\mathrm{Cat}(h(\mathrm{and}, \mathrm{CJ})) = \mathrm{CJ}$, so $\mathrm{PH}(\mathrm{CJ})$ is invariant.
5. To see that $\mathrm{PH}(\mathrm{P}_1)$ is invariant let $\mathrm{Cat}(s) = \mathrm{P}_1$. Let t, u such that $\mathrm{Cat}(t) = \mathrm{NP}$ and $\mathrm{Cat}(u) = \mathrm{P}_2$. Then $\langle(\mathrm{and}, \mathrm{CJ}), s, \mathrm{Merge}(t, u)\rangle \in \mathit{Dom}(\mathrm{Coord})$ hence $\langle h(\mathrm{and}, \mathrm{CJ}), h(s), \mathrm{Merge}(h(t), h(u))\rangle$ is also. But since $\mathrm{PH}(\mathrm{NP})$ and $\mathrm{PH}(\mathrm{P}_2)$ are fixed $\mathrm{Merge}(h(t), h(u))$ has

category P_1, so $h(s)$ must also to coordinate with it. So, by lemma 5.9, $h(\mathrm{PH}(P_1)) = \mathrm{PH}(P_1)$.

6. By similar reasoning any s of category P_0 coordinates with (John laughed,P_0) whose constituents have fixed categories, hence h(John laughed,P_0) has category P_0 so anything we coordinate with it has category P_0, whence $\mathrm{PH}(P_0)$ is fixed.
7. Finally, $\mathrm{PH}(P_1/P_2)$ is invariant. Let s with $\mathrm{Cat}(s) = P_1/P_2$. (Note that there are infinitely many such s). Let h be arbitrary in $\mathrm{Aut}_{\mathrm{Eng}}$ and suppose $\mathrm{Cat}(h(s)) \neq P_1/P_2$. Then $\mathrm{Cat}(h(s)) \in \{P_0, P_1, P_2, \mathrm{NP}, \mathrm{CJ}\}$. But for C in this set, $\mathrm{PH}(C)$ is fixed, hence $\mathrm{Cat}(h^{-1}(h(s))$ is in that set. But $h^{-1}(h(s)) = s$ of category P_1/P_2, a contradiction. So $\mathrm{Cat}(h(s)) = P_1/P_2$, showing that $\mathrm{PH}(P_1/P_2)$ is invariant.

6

A Taste of Mathematical linguistics

Mathematical linguistics usually refers to a field common to mathematics and linguistics where one studies *formal languages*. To see what the field is about, we need some definitions.

We start with set A called the *alphabet*. Recall that the *words on* A are the *sequences* of elements of A. As with all sequences the order of entries matters: *aba* is not the same as *baa*. Repeated entries are permitted. And The empty sequence is a word, written ϵ. The set of all words on A is written A^*, and the set of all non-empty words on A is written A^+. A *language on* A is a set L of words over A. That is, $L \subseteq \mathcal{P}(A^*)$.

(1) Take $A = \{a, b, c\}$ to start. Here are three languages on A.

a. $L_1 = \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc\}$. This is the set of words of *length at most* 2.

b. $L_2 = \{a, aa, aaa, aaaa, \ldots, a^n, \ldots\}$. This is the set of words of length at least 1 consisting of all a's.

c. $L_3 = \{w : w \text{ is a word on } A \text{ but has no repeated letters}\}$. For example $abca \in L_3$, but $abbaca \notin L_3$.

It might seem odd to call a seemingly-random set of sequences a *language*. Here is why this is done. Usually in this chapter, the alphabet A will be a small set of letters, like $\{a, b, c\}$. But it may well be that the alphabet A consists of *words* of a natural language, say English. In this case, the set of English sentences would indeed be a language in our technical sense.

We note some special languages. For any alphabet symbol, say a, $\underline{a}$ is the language with just one word: the one-term sequence x. So $\underline{a} = \{a\}$ for all alphabet symbols a.

We also specify two operations on languages, called $+$ and $\cdot$. Given any two languages on the same alphabet, say L and M, we define

$$\begin{array}{lcl} L + M & = & \text{the union } L \cup M \\ L \cdot M & = & \text{the set of all words obtained by putting one word} \\ & & \text{from } L \text{ in front of one word from } M \end{array}$$

For example: $\underline{a} + \underline{b} = \{a\} \cup \{b\} = \{a, b\}$. And $\underline{a} \cdot \underline{b} = \{a\} \cdot \{b\} = \{ab\}$. Finally, $(\underline{a} + \underline{b}) \cdot \underline{c} = \{a, b\} \cdot \{c\} = \{ac, bc\}$.

The empty language and the language $\{\epsilon\}$ The empty set $\emptyset$ is the set with no elements. It's like an empty bag or box. It's also like zero for union:

$$\emptyset \cup X \quad = \quad X$$

for all sets X.

When it comes to languages, we write 0 for $\emptyset$. We therefore have

$$L + 0 \quad = \quad L \quad = \quad 0 + L$$

for all languages L.

But $\{\epsilon\}$ is different from $\emptyset$. The reason is simple: $\{\epsilon\}$ has an element, while $\emptyset$ has no elements. As a short notation, we write 1 for $\{\epsilon\}$.

The star operation $*$ **on languages** For any language L, L^* is the set of words which we can make by putting together zero or more elements of L, one after the other.

For example, let $L = \{a, bb\}$. Then

$$\{a, bb\}^* \quad = \quad \{\epsilon, a, bb, abb, bba, bbbba, bbaaabb, bbabbaabbabb, \ldots\}.$$

In words, $\{a, bb\}^*$ will have all the words consisting of only as and bs, and with the property that the consecutive strings of b's always have even length.

Exercise 6.1. Let $L = \underline{a}^*$.

a. Write L out in list notation.
b. Check that $L + L = L$.
c. What language is $L \cdot L$? Is it the same as $L + L$?

Exercise 6.2. Try writing some languages in list notation, then if possible describing them in English.

a. $(\underline{a} + \underline{b})^*$
b. $\underline{a} \cdot (\underline{a} + \underline{b})^*$
c. $(\underline{a} + \underline{b}) \cdot (\underline{a} + \underline{b})^*$
d. $(\underline{a} + \underline{b})^* \cdot \underline{c}$
e. $(\underline{a}^*)^*$

Exercise 6.3. Let $A = \{a, b, c\}$, and go the other way, producing regular expressions from descriptions.

a. The language of all words on A which begin with b.
b. The language of all words on A which have even length.
c. The language of all words on A which have length at least 2.
d. The language of all words on A which do not have two a's in a row. (This last one is harder than the others!)

Exercise 6.4. What are $0 \cdot L$ and $1 \cdot L$? What are $0 + L$ and $1 + L$? What is 0^*? What is 1^*?

Back to *the mother of the president* As a way of bringing our rather abstract discussion back to linguistic reality, we want to make the case that the kinds of repetitive phenomena in language that we saw in Chapter 1 can mostly be handled by regular languages. Let

$$A \quad = \quad \{\text{the, mother, of, president}\}.$$

This is an alphabet of four "letters", the letters being in this case words of English. Then a few simple calculations (which we hope you will work out for yourself) show that

$$(\underline{\text{the}} \cdot \underline{\text{mother}} \cdot \underline{\text{of}})^* \cdot (\underline{\text{the}} \cdot \underline{\text{president}})$$

is a set we saw before, namely

$$\{\epsilon, \text{the president}, \text{the mother of the president}, \ldots\}$$

So part of our point behind regular languages is that they cover the basic phenomena that make language infinite. At the same time, they lack any notion of *constituency* that played such a big role in Chapter 4.

6.1 Regular Expressions and Languages

Fix an alphabet A. A *regular expression* is something you can write out of symbols a, for $a \in A$; fresh symbols 0 and 1; binary function symbols $+$ and $\cdot$, and a unary function symbol $*$. For examples, we mention a, $a+b$, $b+a$ (this is a different expression from $a+b$), a^*, and $b+(a^* \cdot (c^*)^*)$. We use letters like r and s to denote regular expressions, and we let R be the set of all regular expressions on our alphabet A.

We define a function $\mathcal{L} : R \to \mathcal{P}(A^*)$, giving for each regular expression r its *denotation language* $\mathcal{L}(r)$. This function is defined as follows:

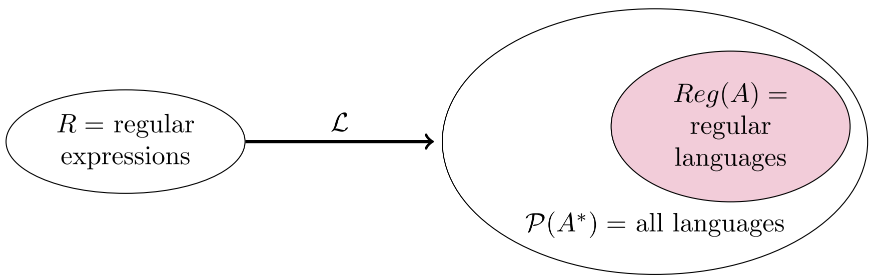
$$\begin{array}{lcl}
\mathcal{L}(a) & = & \underline{a} \\
\mathcal{L}(r+s) & = & \mathcal{L}(r) + \mathcal{L}(s) \\
\mathcal{L}(r \cdot s) & = & \mathcal{L}(r) \cdot \mathcal{L}(s) \\
\mathcal{L}(r^*) & = & (\mathcal{L}(r))^* \\
\mathcal{L}(0) & = & 0 \ (= \emptyset) \\
\mathcal{L}(1) & = & 1 \ (= \{\epsilon\})
\end{array}$$

Here are some examples with $A = \{a, b, c\}$:

regular expression r	its associated language $\mathcal{L}(r)$
$\underline{a}$	$\{a\}$
$\underline{a}+\underline{b}$	$\{a, b\}$
$(\underline{a}+\underline{b})\cdot\underline{a}$	$\{aa, ba\}$
$(\underline{a}+\underline{b})^*$	$\{\epsilon, a, b, aa, ab, ba, bb, \ldots\}$
$\underline{c}\cdot(\underline{a}+\underline{b})^*$	$\{c, ca, cb, caa, cab, cba, cbb, \ldots\}$

Each regular expression *denotes* a language, as we have seen. The set $Reg(A)$ of *regular languages on* A are defined to be the set of all denotations of regular expressions:

$$Reg(A) \quad = \quad \{\mathcal{L}(r) : r \in R\}.$$



$\mathcal{L}$ is a function from regular expressions to the set of languages. The regular languages are defined as the image set of this function $\mathcal{L}$. $\mathcal{L}$ is not one-to-one.

Exercise 6.5. Let $A = \{\text{John}, \text{Mary}, \text{Sam}, \text{knows}, \text{that}\}$. Some of the words on A are proper sentences in English, and some are not. Write a regular expression that denotes the set of *all* words on A that *are* English sentences.

Often one writes $r = s$ when $\mathcal{L}(r) = \mathcal{L}(s)$. For example, one would usually write

$$a+b \quad = \quad b+a,$$

since

$$\mathcal{L}(a+b) \quad = \quad \underline{a}+\underline{b} \quad = \quad \{a,b\} \quad = \quad \underline{b}+\underline{a} \quad = \quad \mathcal{L}(b+a).$$

But it is important to see that there is a difference between an expression r and the language $\mathcal{L}(r)$ it denotes.

Why are we interested in regular expressions? First, regular expressions are a very simple form of grammar. There is the tantalizing possibility that one could take

$$A \quad = \quad \text{the set of words of English}$$

and write a single (huge) regular expression that would give the set of all English sentences.

The notion of regular expressions also has a practical value: grammar checkers on a computer are based on regular expressions.

We are also interested in them because they highlight *laws of abstract algebra* such as

$$\begin{array}{lcl} L+M & = & M+L \\ L\cdot(M_1+M_2) & = & (L\cdot M_1)+(L\cdot M_2) \\ (L^*)^* & = & L^* \end{array}$$

that we'll study in greater depth later on.

The operation $^+$ We have seen the * operation before, and we want a variant of it called $^+$. $\mathcal{L}(r^*)$ is all sequences of words in $\mathcal{L}(r)$, smashed together to give one single word, *including* the empty word ϵ.

In contrast, $\mathcal{L}(r^+)$ is all sequences of words in $\mathcal{L}(r)$, smashed together to give one single word, *not including* ϵ.

regular expression r	its associated language $\mathcal{L}(r)$
$\underline{a}^*$	$\{\epsilon, a, aa, aaa, \ldots\}$
$\underline{a}^+$	$\{a, aa, aaa, \ldots\}$
$(\underline{a}+\underline{b})^+$	$\{a, b, aa, ab, ba, bb, \ldots\}$
$\underline{c}\cdot(\underline{a}+\underline{b})^+$	$\{ca, cb, caa, cab, cba, cbb, \ldots\}$

An interesting fact If we have a regular expression r, and it just so happens that $\epsilon \notin \mathcal{L}(r)$, then we can find a different regular expression *s that uses $^+$ instead of ** such that $\mathcal{L}(s) = \mathcal{L}(r)$.

For example: let r be $(\underline{a}^*\cdot\underline{b})+(\underline{a}\cdot\underline{b}^*)$. Then

$$\mathcal{L}(r) \quad = \quad \{a, b, ab, aab, aaab, \ldots, abb, abbb, \ldots\}.$$

It is any non-empty sequence of a's followed by any non-empty sequence of b's. The point is that it is also $\mathcal{L}(s)$, where

$$s \quad = \quad (\underline{a}^+\cdot\underline{b})+(\underline{a}\cdot\underline{b}^+)+(\underline{a}+\underline{b}),$$

and this last expression used $^+$ instead of *.

Exercise 6.6. let r be $(\underline{a}^*\cdot\underline{b})+(\underline{a}\cdot\underline{b}^*)$. What is the language $\mathcal{L}(r)$? Can you find an expression s using $^+$ instead of * such that $\mathcal{L}(s) = \mathcal{L}(r)$?

6.2 Simple Categorial Grammars

What we want to do at this point is to take a regular language without ϵ and write a categorial grammar for it. As an first example of what we are trying to do, consider $\underline{a}^+$ as a language over the alphabet $A = \{a, b, c\}$. We take a basic category S. This will be our only basic category. We take as the lexicon

$$(a, \mathrm{S}), (a, \mathrm{S}/\mathrm{S}).$$

(Actually, we mean the *set* with these two elements, but frequently we omit the set braces.) And we take S to be our top-level category. So now we have a categorial grammar that we'll call $\mathcal{G}(a^+)$. We can check that $\mathcal{L}(\mathcal{G}) = \underline{a}^+$.

Here is a second example, a grammar for $\underline{a}^+ + \underline{b}^+$. We take a basic category S, and also basic categories A and B. (We have complete freedom over what basic categories to take. We usually use S as our "top level" category, but this is not strictly required. And we can name our categories A and B; they do not have to be named N, DP, etc.) Here is our lexicon:

(a, A)	(b, B)
$(a, A/A)$	$(b, B/B)$
(a, S)	(b, S)
$(a, S/A)$	$(b, S/B)$

Let's call this grammar $\mathcal{G}_2$. Try to *parse* aa and bbb as Ss in this grammar $\mathcal{G}_2$. Also let's convince ourself that ab is not parsable as S in $\mathcal{G}_2$. The important point is that $\mathcal{L}(\mathcal{G}_2) = \underline{a}^+ + \underline{b}^+$.

We are driving towards the following general fact:

Theorem 6.1. *Let A be an alphabet. Let r be a regular expression over A built from*

- $\underline{a}$*, where* $a \in A$.
- *the operations* $+$, $\cdot$, *and* $^+$ *(but not* **).*

Then there is a categorial grammar $\mathcal{G}(r)$ such that

- $\mathcal{L}(\mathcal{G}(r)) = \mathcal{L}(r)$. *In words, the set of words over A which can be parsed in $\mathcal{G}(r)$ as Ss is exactly $\mathcal{L}(r)$, where S is the top-level symbol of $\mathcal{G}(r)$.*
- *All categories in the lexicon of $\mathcal{G}(r)$ are either basic categories, or are of the form X/Y for basic categories X and Y.*

The proof actually gives a method of coming up with $\mathcal{G}(r)$ from r. We go *bottom-up*.

First, we start with the simplest case, when r is $\underline{a}$ for some $a \in A$. We take a basic category S, a lexicon $\{(a, S)\}$, and declare S to be the top-level symbol. This gives a grammar $\mathcal{G}$. Clearly $\mathcal{L}(\mathcal{G}(r)) = \{a\} = \mathcal{L}(r)$.

Combining grammars $\mathcal{G}(r)$ and $\mathcal{G}(s)$ to get a grammar $\mathcal{G}(r + s)$ Suppose we have $\mathcal{G}(r)$ for r, and also $\mathcal{G}(s)$ for s. Here is how to combine these to get a grammar $\mathcal{G}(r + s)$ for $r + s$.

- Re-name all the basic categories in $\mathcal{G}(r)$ and $\mathcal{G}(s)$ so that the grammars have no categories in common. Make sure that S is *not* a basic category in either grammar.
- Call the top-level symbol of $\mathcal{G}(r)$ T, and call U the top-level symbol of $\mathcal{G}(s)$.
- Put the lexicons together in one big set.
- Add entries to the lexicon: If it has (a, T), add (a, S). If it has $(a, T/X)$, add $(a, S/X)$. If it has (a, U), add (a, S). $(a, U/X)$, add $(a, S/X)$.
- Declare S to be the top-level symbol of the new grammar.

Combining grammars $\mathcal{G}(r)$ and $\mathcal{G}(s)$ to get a grammar $\mathcal{G}(r \cdot s)$ Suppose we have $\mathcal{G}(r)$ for r, and also $\mathcal{G}(s)$ for s. Here is how to combine these to get a grammar $\mathcal{G}(r \cdot s)$ for $r \cdot s$.

- Re-name all the basic categories in $\mathcal{G}(r)$ and $\mathcal{G}(s)$ so that the grammars have no categories in common.
- Let S be the top-level symbol in $\mathcal{G}(r)$.
- Let T be the top-level symbol in $\mathcal{G}(s)$.
- Put the lexicons together in one big set.
- Make *changes* to the lexicon: If X is a basic category in $G(r)$ and the lexicon has (a, X), then change this entry to $(a, X/T)$.
- Declare S to be the top-level symbol of the new grammar.

Changing $\mathcal{G}(r)$ to get $\mathcal{G}(r^+)$ Suppose we have $\mathcal{G}(r)$ for r. Here is how to modify it to get a grammar $\mathcal{G}(r^+)$ for r^+.

- Let S be the top-level symbol in $\mathcal{G}(r)$.
- Make a few *additions* to the lexicon: If X is a basic category in $G(r)$ and the lexicon has (a, X), then *add* $(a, X/S)$.
- Declare S to be the top-level symbol of the new grammar.

An example: a grammar for $(\underline{a} \cdot \underline{b})^+ + \underline{c}^+$. We build our grammar from the bottom-up. First, we know how to construct grammars $\mathcal{G}(\underline{a})$, $\mathcal{G}(\underline{b})$, and $\mathcal{G}(\underline{c})$. Our last point tells us how to get a grammar $\mathcal{G}(\underline{c}^+)$: we would take a basic category S, a lexicon $\{(c : S), (c : S/S)\}$, and of course use S as the top-level symbol.

Second, we combine $\mathcal{G}(\underline{a})$ and $\mathcal{G}(\underline{b})$ as follows. Rename S to T in $\mathcal{G}(\underline{b})$, change the lexicon entry $(a : S)$ to $(a : S/T)$, and put the lexicons together. We get

$$(a : S/T), (b : T)$$

with S as the top-level symbol.

Third, to get $\mathcal{G}((\underline{a} \cdot \underline{b})^+)$, add the entry $(a : T/S)$ to the last lexicon.

Finally, we make additions for the overall $^+$ operation. We get the grammar below:

(2)

$$\begin{array}{ll} (a, T/X) & (c, Y) \\ (a, S/X) & (c, Y/Y) \\ (b, X) & (c, S) \\ (b, X/T) & (c, S/Y) \end{array}$$

Once again, our top-level symbol is S.

Review and a definition Given a regular expression r, we found a categorial grammar $\mathcal{G}(r)$ which generated $\mathcal{L}(r)$. In fact, $\mathcal{G}(r)$ was "simple".

Definition 6.1. A categorial grammar is *simple* if all the categories in the lexicon are either basic categories, or else X/Y, where X and Y are basic categories. An *SCG* is a simple categorial grammar with a specified choice of a "top-level" category.

Given a regular expression r, we found an SCG $\mathcal{G}(r)$ whose language is $\mathcal{L}(r)$. That is, $\mathcal{L}(\mathcal{G}(r)) = \mathcal{L}(r)$.

Exercise 6.7. Although we indicated the steps in constructing $\mathcal{G}(r)$ from r, we did not actually prove that $\mathcal{L}(\mathcal{G}(r)) = \mathcal{L}(r)$. If you know about induction, prove that indeed $\mathcal{L}(\mathcal{G}(r)) = \mathcal{L}(r)$.

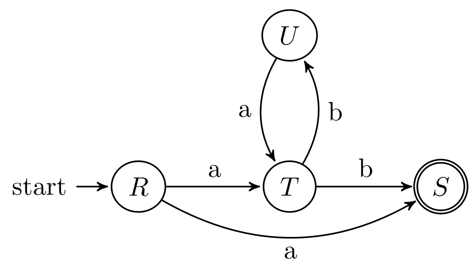
Our next goal is to show that if someone gives us an SCG, then its language *must be* a regular language without ϵ. That is, we'll take an SCG $\mathcal{G}$ and (after many steps) come up with a regular expression r using $+$ with the property that $\mathcal{L}(r) = \mathcal{L}(\mathcal{G})$.

To solve this kind of problem in an insightful and general way, we make yet another digression, to the topic of *automata*.

6.3 Finite-state Automata

A *finite-state automaton* is a mathematical object that is easy to grasp as a picture and then to remember the definition afterwards. For example, here is a picture of an automaton:

(3)



We'll call this $\mathcal{A}$. It has four *states*: R, S, T, and U. R is called the *start state*, indicated by the arrow. S is the *accepting state*, indicated by the double circle.

We want to take words on our alphabet and ask whether or not it's possible to read the letters beginning at the start state and ending at the accepting state of $\mathcal{A}$, following the arrows. We can make a chart to give some examples:

word	accepted?
a	$\surd$
b	$\times$
aa	$\times$
ab	$\surd$
ba	$\times$
bb	$\times$

word	accepted?
aba	$\times$
abb	$\times$
$abaa$	$\times$
$abab$	$\surd$
$ababab$	$\surd$
$abababab$	$\surd$

To be accepted, a word only needs one path. It may take some trial-and-error to see if a word actually is accepted by a given automaton, or not. The *language of the automaton* $\mathcal{A}$ is the set of words which *are* accepted by $\mathcal{A}$ in the sense above.

For the automaton in (3), the language is

$$\begin{aligned} & \{a, ab, abab, ababab, abababab, \ldots\} \\ = \quad & \underline{a} + (\underline{a} \cdot \underline{b})^+ \end{aligned}$$

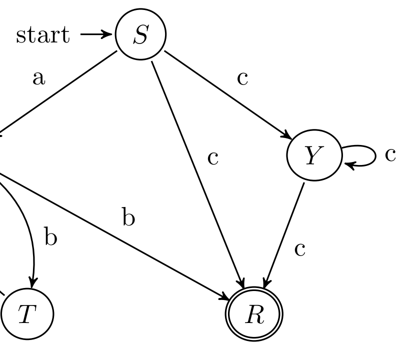
Converting categorial grammars to automata Here is how to take an SCG $\mathcal{G}$ and associate an automaton $\mathcal{A}(\mathcal{G})$ to it:

- The states of automaton $\mathcal{A}(\mathcal{G})$ are the basic categories of the grammar $\mathcal{G}$, plus a new state that we'll call R.
- The top-level state of the $\mathcal{G}$ is the initial state of $\mathcal{A}(\mathcal{G})$.
- If the lexicon has (a, X), then add an arrow from X to R labeled by a.
- If the lexicon has $(a, X/Y)$, then add an arrow from X to Y labeled by a.

► The new state R is the accepting state of $\mathcal{A}(\mathcal{G})$.

As an example, consider our grammar for $(\underline{a}\cdot\underline{b})^+ + \underline{c}^+$ from (2) above. The automaton for it is shown below:

(4)



Proposition 6.2. *In the notation above, the language of the automaton is the one we started with:*

$$\mathcal{L}(\mathcal{G}) \quad = \quad \mathcal{L}(\mathcal{A}(\mathcal{G})).$$

We are missing a piece, the function marked "solution." This associates a regular expression to each automaton. To get this, we shall

► take an automaton

► turn it into an algebra problem, but different from high school algebra

► solve the problem, obtaining a regular expression.

For example, look back at the automaton in (4) above. Here is the associated system of equations:

(5)

$$\begin{array}{lcl} X_R & = & 1 \\ X_Y & = & cX_Y + cX_R \\ X_U & = & bX_T + bX_R \\ X_T & = & aX_U \\ X_S & = & aX_U + cX_R + cX_Y \end{array}$$

Here and below, X_S and X_T are *variables ranging over languages.* We'll have one variable for each state of the automaton, and one equation per variable. We got each equation X_S by looking at the *outgoing edges from* S. We changed the state X to U to avoid the terrible notation X_X.

We get a *system of equations* in the variables. The idea is that X_S is the set of words w which when read in at state S could lead by some

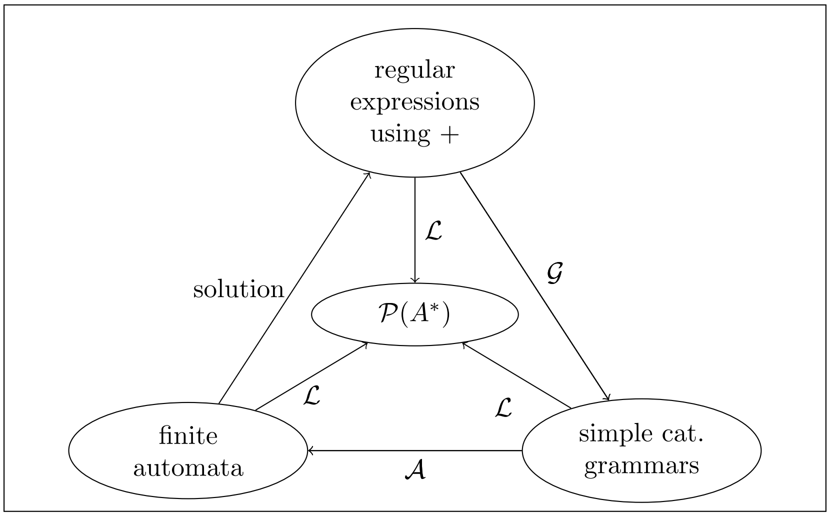


FIGURE 8 The formalisms in this section. The functions $\mathcal{L}$ all map into the set $\mathcal{P}(A^*)$ of all languages over A. But the image sets of the three $\mathcal{L}$ functions are the same, the set of regular languages without ϵ.

path to the accepting state R. (And similarly for the other variables.) So $X_R = 1$, since $1 = \{\epsilon\}$, and this is the only word which, when started in R, leads back to R itself. And the equation for X_S means that a word w could be read in at state S to get to R if and only if

$w = av$, where v can be read in at U
or $w = cv$, where v can be read in at R
or $w = cv$, where v can be read in at Y

Solving a system Here is the most basic observation: To solve $X = vX+w$, where w is a word, and w is any expression that does not involve X. (Note: w might involve other variables besides X.) Then the solution of this equation is

$$X = v^* \cdot w$$

That is, the only language X such that $X = vX + w$ is X^*. But to use this key fact, you have to have a variable alone on the left side of an equation, and it has to occur on the right side.

In addition, you can do things similar to what you do in algebra,

► substitute partially done work

► use laws of algebra to simplify things

A worked example We return to (5), shown again below.

$$\begin{array}{lll} X_R & = & 1 \\ X_Y & = & cX_Y + cX_R \\ X_U & = & bX_T + bX_R \\ X_T & = & aX_U \\ X_S & = & aX_U + cX_R + cX_Y \end{array}$$

X_R is already solved. So we go back and rewrite a few of the equations:

$$\begin{array}{lllll} X_R & = & 1 & & \\ X_Y & = & cX_Y + c{\cdot}1 & = & cX_Y + c \\ X_U & = & bX_T + b{\cdot}1 & = & bX_T + b \\ X_T & = & aX_U & & \\ X_S & = & aX_U + c{\cdot}1 + cX_Y & = & aX_U + c + cX_Y \end{array}$$

We see first that $X_Y = c^* \cdot c = c^+$. We can plug this in to the equation for X_S:

$$\begin{array}{lll} X_R & = & 1 \\ X_Y & = & c^+ \\ X_U & = & bX_T + b \\ X_T & = & aX_U \\ X_S & = & aX_U + c + c \cdot c^+ \end{array}$$

We then plug the right-hand side of the X_T equation in for X_T in the right-hand side of the X_U equation:

$$\begin{array}{lllll} X_R & = & 1 & & \\ X_Y & = & c^+ & & \\ X_U & = & b(aX_U) + b & = & (ba)X_U + b \\ X_T & = & aX_U & & \\ X_S & = & aX_U + c + c(c^+) & & \end{array}$$

We solve for X_U in one step; it's $(ba)^*b$. So we get

$$\begin{array}{lll} X_R & = & 1 \\ X_Y & = & c^+ \\ X_U & = & (ba)^*b \\ X_T & = & a(ba)^*b \\ X_S & = & a(ba)^*b + c + c(c^+) \end{array}$$

And then we get

$$\begin{array}{lll} X_S & = & a(ba)^+b + c + (c \cdot c^+) \\ & = & a(ba)^*b + c^+ \end{array}$$

The overall variable that we are interested in is the *start state* of the automaton. In this case, it is X_S. So the *overall answer* for this automaton $\mathcal{A}$ is

$$\text{solution}(\mathcal{A}) \quad = \quad a(ba)^*b + c^+.$$

An equivalent (and better) way to write this answer is to notice that $a(ba)^*b = (ab)^+$. So the *overall solution* for this automaton $\mathcal{A}$ is

$$\text{solution}(\mathcal{A}) \quad = \quad (ab)^+ + c^+.$$

This is not surprising, since we got $\mathcal{A}$ from a SCG for $(ab)^+ + c^+$!

A summary of what we have done The progression of ideas is

$$\text{Language} \Longrightarrow \text{SCG} \Longrightarrow \text{automaton} \Longrightarrow \text{system} \Longrightarrow \text{solution}$$

We have indicated this in more detail in Figure 8 above. We know now that all of the $\mathcal{L}$ functions have the same image set, the set of regular languages without ϵ. In particular, *the language of a simple categorial grammar is regular.*

Here is the reasoning in more detail. Let $\mathcal{G}$ be a SCG. Then $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{A}(\mathcal{G}))$. And $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\text{solution}(\mathcal{A}(\mathcal{G})))$. Now solution$(\mathcal{A}(\mathcal{G})))$ is a regular expression using $^+$, because our solutions always are regular expressions using $^+$, and so $\mathcal{L}(\textit{solution}(\mathcal{A}(\mathcal{G})))$ *is a regular language without* ϵ.

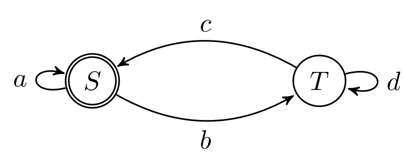
Therefore, $\mathcal{L}(\mathcal{G})$ is a regular language without ϵ.

Overall In this section, we have explored several ways of defining languages:

- using regular expressions (with $^+$)
- using categorial grammars, and (especially) simple ones
- using automata

By doing some significant work, we showed that all of these are *equivalent*: they are different ways of representing the same set of languages, the regular languages without ϵ.

Exercise 6.8. Consider the automaton below.



Write the system of equations, and solve it. In this way, you'll get $\mathcal{L}(\mathcal{A})$, the language of this automaton.

Exercises 6.9-6.13 have to do with the language $\mathcal{L}(r)$, where

$$r \quad = \quad (\underline{a} \cdot (\underline{b}^+ + \underline{c}))^+.$$

Exercise 6.9. Find a simple categorial grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G}) = \mathcal{L}(r)$. That is, you should come up with a set of basic categories, find a lexicon for a, b, and c, and finally say what your top-level category is.

Exercise 6.10. Take the grammar that you found in Exercise 6.9, and turn it into an automaton. Be sure to indicate which is your "start state", and also which state is your accepting state.

Exercise 6.11. Take the automaton from the Exercise 6.10, and find the corresponding system of equations.

Exercise 6.12. Take the system of equations from the Exercise 6.11, and find a regular expression for the set of words accepted when we read them in, beginning in the start state of the automaton from problem 6.10. [This means that you solve the system using the version of algebra presented in the notes.]

Exercise 6.13. If you did all the previous problems correctly, your answer to Exercise 6.12 should be the original regular expression, $(\underline{a} \cdot (\underline{b}^+ + \underline{c}))^+$. Check that it is. [As a hint, you might find it useful to note that for all words u and v,

$$(u \cdot v)^* \cdot u \quad = \quad u \cdot (v \cdot u)^*.$$

That is, use a special case of this, with a well-chosen u and v.]

Exercise 6.14. The *reversal* of any language L is the set of elements in L written backwards. For example, the reversal of $ab = \{ab\}$ is ba. Write a regular expression for the *reversal* of $\mathcal{L}(r)$, where r again is $(\underline{a} \cdot (\underline{b}^+ + \underline{c}))^+$.

Exercise 6.15. Prove that the reversal of every regular language is regular.

6.4 More About Regular Languages

Our goal in this chapter is not to present an entire course on formal language theory, only to touch on the subject. We have done most of the background work on regular language The section begins with a question: Are CGs more powerful than SCGs?

The answer turns on a language called $a^n b^n$. This language is the set

$$\begin{aligned} & \{a^n b^n : n \geq 1\} \\ = \quad & \{ab, aabb, aaabbb, \ldots, a^n b^n, \ldots\} \end{aligned}$$

(The name is a little unfortunate, since usually "n" would be for a *fixed* number.) We'll show that $a^n b^n$ is not a regular language, and then use that to argue that SCGs are not adequate for natural languages.

We are going to assume that $a^n b^n$ *is* regular, and then get a contradiction. Since regular languages can be presented in three ways, we have presentations of $a^n b^n$ in all three. We thus would have a finite automaton, say $\mathcal{A}$, whose language $\mathcal{L}(\mathcal{A})$ was exactly $a^n b^n$.

(Recall that $\mathcal{L}(\mathcal{A})$ is the set of words which can be read in from the start state of $\mathcal{A}$ and have a path leading to the accepting state of $\mathcal{A}$.)

$\mathcal{A}$ is finite, and so it has some number of states. Let's say that M is that number. One of the words in $a^n b^n$ is $a^{M+1} b^{M+1}$. When we read $a^{M+1} b^{M+1}$ in to $\mathcal{A}$, we begin at the start state, follow some path or other, and somehow end up at the accepting state. But thos word is longer than the number of states, so there must be a *loop* somewhere. In fact, there has to be a loop while reading the a's. Let's say that when after ith and jth a's were read in, the automaton was in the very same state. Call this state X, and call the accepting state Y.

letter	a	a	$\cdots$	a	$\cdots$	a	$\cdots$	a	b	b	$\cdots$	b
state	start			X	$\cdots$	X	$\cdots$					Y

Since we have a loop in the states, we can *repeat* the blue a's, except for the last one:

letter	a	a	$\cdots$	a	$\cdots$	a	$\cdots$	a	$\cdots$	a	b	b	$\cdots$	b
state	start			X	$\cdots$	X	$\cdots$	X	$\cdots$					Y

letter	a	a	$\cdots$	a	$\cdots$	a	$\cdots$	a	$\cdots$	a	b	b	$\cdots$	b
state	start			X	$\cdots$	X	$\cdots$	X	$\cdots$					Y

What word is read in? How many a's and how many b's do you see? It's

$$a^{M+j-i+1} b^{M+1}.$$

Since $j > i$, this word is *not* in the language $a^n b^n$. But it *is* accepted by $\mathcal{A}$. So we have a contradiction! And our proof is done!

(The point, informally, is that since our original word was read in using a loop in the states, we can go around the loop twice. The automaton has no memory, and so it must accept the longer word.)

We just saw that $a^n b^n$ cannot be the language of any finite automaton, and it follows that we cannot write an SCG for it. But if we allow ourselves the full power of categorial grammar, we *can* write a grammar for it. We take basic categories X and S. The lexicon is

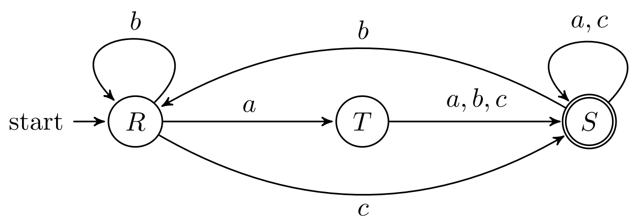
$$(a,\ S/X) \qquad (b,\ S\backslash X) \qquad (b,\ X)$$

The top-level symbol is S.

We asked earlier: Are CGs more powerful than SCGs? We now know that the answer to this is Yes. We next want to ask the question of whether SCGs are powerful enough to handle a natural language like English. The answer requires some more theory, so we turn to that next.

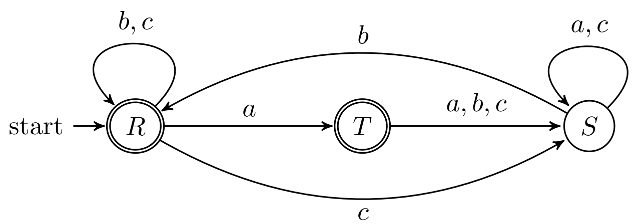
Definition 6.2. An automaton is *deterministic* if for every state X and every alphabet symbol a, there is one and only one arrow out of S labeled with a.

As an example, consider



(6)

Automata with more than one accepting state So far this chapter, we only dealt with automata with *one* accepting state. But automata can have more than one accepting state.



(7)

We say that a word w is *accepted* in $\mathcal{A}$ if w can be read in from the start state, following the arrows as we read it in, and end up in *some accepting state or other*. We still get a *regular language*.

Exercise 6.16. Find the language of the automaton just above. To do this, write the equations of the automaton down, and solve them. Then take the solutions of X_R and X_T (these are regular expressions), and take their sum.

Exercise 6.17. Use the idea in the last exercise to justify the following claim: if $\mathcal{A}$ is an automaton with more than one accepting state, then $\mathcal{L}(\mathcal{A})$ is a regular language.

Now above we drew two automata; see (6) and (6). The relation between the two of these is that the accepting states of the first are the non-accepting states of the second (and vice-versa). The next result clarifies the situation.

Proposition 6.3. *Suppose that $\mathcal{A}$ is a* deterministic *automaton. Let $\mathcal{B}$ be the same automaton, but when we switch the accepting and non-accepting states. Then $\mathcal{L}(\mathcal{B})$ is the* complement *of $\mathcal{L}(\mathcal{A})$:*

$$\mathcal{L}(\mathcal{B}) \quad = \quad -\mathcal{L}(\mathcal{A}).$$

Proposition 6.4. *For* any *automaton $\mathcal{A}$ (even a non-deterministic $\mathcal{A}$), there is a deterministic $\mathcal{B}$ with the same language: $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.*

This takes a special proof that I'll outline in homework. It's definitely not easy or obvious.

Putting the two facts together:

Theorem 6.5. *The complement of any regular language L is also regular.*

Proof. Let L be regular. This means that we have $\mathcal{A}$ so that $L = \mathcal{L}(\mathcal{A})$. By Proposition 6.4, let $\mathcal{B}$ be deterministic and have the same language as $\mathcal{A}$. We swap accepting and non-accepting states in $\mathcal{B}$ to get an automaton that we call CC. By Proposition 6.3, $\mathcal{L}(CC)$ is the complement of $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B}) = L$. And $\mathcal{L}(CC)$ is regular, since it is the language of some automaton. So the complement $-L$ of L is regular. □

Corollary 6.6. *If L and M are regular, so is $L \cap M$.*

Proof. Let L and M be regular. We prove that $L \cap M$ is also regular. By facts earlier this semester (de Morgan's law),

$$L \cap M \quad = \quad -((-L) + (-M)).$$

Now $-L$ and $-M$ are regular, by what we just did. And the union of two regular languages is also regular: just add regular expressions for them. So $(-L) + (-M)$ is regular. And then using complement one more time, we see that $-((-L) + (-M))$ is regular. □

6.5 An Argument Why English is Not Regular

We have studied regular languages to such a large extent that you might think that we felt they are particularly useful in the study of language. The opposite is the case: we present an argument below to the effect that natural languages are *not* regular. So in a sense, regular languages are a diversion, almost a "straw man". But the reason that we have been interested in them is that they present a clear model that we can easily work with. (In addition, they are very useful in areas such as computational linguistics.) We shall go beyond the regular languages later in this chapter, but now we turn to our argument.

Suppose towards a contraction that English were regular. Consider L defined by

$$L \quad = \quad \text{English} \cap \{\text{the}, \text{boy}, \text{girl}, \text{saw}, \text{ran}\}^+$$

Since the intersection of two regular languages is regular, this language L must also be regular.

Here are some sentences in L:

the boy ran
the boy the girl saw ran
the girl the girl saw ran
the boy the girl *the girl saw* saw ran
the boy the girl *the girl* *the girl saw* *saw* saw ran

We know that nobody can possibly understand this last sentence, but there are reasons why linguists want to take it to be part of *ideal English*.

And the list goes on.

The main things about L are that

(1)No sentence has two "the"s in a row.

(2)No sentence has two "boy"s in a row, or two "girl"s.

(3)the sentence has exactly as many nouns as "see"s.

Now we repeat the proof that $a^n b^n$ is not regular. We know now that L is regular, so for some automaton, say $\mathcal{A}$, $L = \mathcal{L}(\mathcal{A})$. Let M be the number of states in $\mathcal{A}$.

Consider

$$\text{the girl the girl the girl} \cdots \text{ saw saw} \cdots \text{ saw ran}$$

We would have one more "girl" than "saw". But by considering loops in $\mathcal{A}$ again, this automaton must also accept a string of words with more stuff in the first part of the sentence.

As a result $\mathcal{A}$ would accept a string that violated one of our points (1)–(3) above. And this string would definitely *not* be an English sentence, hence not in L. So this proves that $\mathcal{L}(\mathcal{A}) \neq L$.

6.6 Context-Free Grammars and Languages

At this point, we have some indication that regular expressions and SCGs are not adequate for the description of syntax. We should mention that even if they were adequate, linguists would not be terribly happy with formalisms like regular expressions, automata, and SCGs. The reason is that they do not give a direct handle on the notion of constituency. So even if the grammars "worked out" in the sense of being able to correctly model interesting syntactic phenomena, they would still not be "right" in the sense that their study would lead to interesting questions for syntax. We next turn to a more powerful formalism, the *context-free grammars* that comes much closer to being both powerful enough to represent interesting syntactic phenomena and also can give constituency structures (trees) that are much closer to what a syntacticians use.

We start with a set NT of *non-terminals* and also a set T of *terminals*. For example, the terminals can be words or letters. And the non-terminals can be grammatical categories. Usually we write the terminals with lower case letters a, b, $\ldots$, and the non-terminals are written with upper case X, Y, S, A, B, $\ldots$

A *CFG rule* is an expression

$$X \to w,$$

where $X \in NT$ and $w \in (T \cup NT)^+$. A *CFG* is a finite set of CFG rules, and a specification of one of the non-terminals to be the *top-level* non-terminal. We have yet to say what CFG's *do*, and how they generate languages.

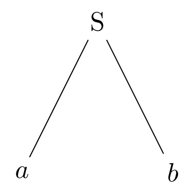
Here is a basic example of a context-free Grammar (CFG). We take $T = \{a, b\}$, and $NT = \{S\}$. We have two rules: $S \to aSb$ and $S \to ab$. We also state that the top-level symbol is S. This is a context-free grammar that we'll call G.

Parse trees A parse tree for a CFG G is a tree T with a few important properties:

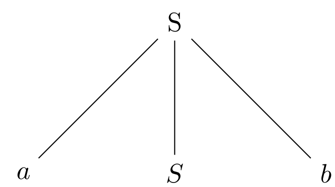
a. The tree is labeled with elements of $NT \cup T$.
b. The leaves must be labeled with terminal symbols (T).
c. The non-leaves must be labeled with non-terminal symbols (NT).
d. The root of T must be labeled with the top-level non-terminal.
e. Each non-leaf must match some rule in the grammar G.
f. T must be finite; it cannot go on forever.

The leaf sequence of a parse tree T is called the *yield* of T. The *language of the grammar* G is the set of all yields of all parse trees of G. Each CFG G gives us a language $\mathcal{L}(G)$. The *context-free languages* are all the languages $\mathcal{L}(G)$, where G can be any CFG.

Compare this with the *regular languages*. For example, let's look at the parse trees in our example grammar G.

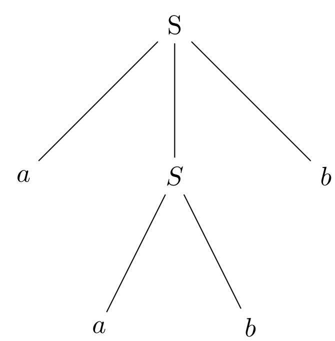


This is a good parse tree for our grammar.



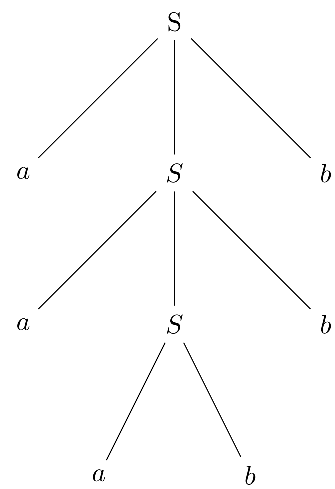
This is not quite a parse tree, since one of the leaves is not labeled with a terminal symbol.

Here is the second-smallest parse tree in the grammar.



Its yield is *aabb*.

What is the third-smallest parse tree?



The yield is *aaabbb*.

All of the parse trees look like the three that we have seen. The set of words which are yields of parse trees is

$$\{ab, aabb, aaabbb, \ldots, a^n b^n, \ldots\}$$

This is the *language of this grammar*. We write this as $\mathcal{L}(G)$.

Exercise 6.18. Write grammars for some languages.

a. a^+
b. a^+b^+
c. $(a+b)^+(c+d)^+$
d. $((a+b)^+(c+d)^+)^+$

Exercise 6.19. Prove that every regular language without ϵ is a context-free language. There are two ways that you can go about this; you have your choice. You can convert regular expressions into context-free grammars, or you can convert SCG's. Either way, you should give some indication of *how* your method works (by giving an example or two), and also *why* it works.

A trickier example Let $T = \{(,), a, b, c, +, \cdot, ^*\}$. (Note that we have the parentheses as terminal symbols (!) and also the symbols $+$, $\cdot$, and *. Let's write a grammar G whose language $\mathcal{L}(G)$ is the *set of all regular expressions*: So we want $\mathcal{L}(G) = \{a, b, (a+b), (a \cdot b), (a+b)^*, \ldots\}$.

Note that we want $\mathcal{L}(G)$ to have very "official looking" expressions, with more parentheses than we usually would see. We take one terminal symbol, S.

Of course S will be the top-level symbol of the grammar.
For the rules, we take

$$
\begin{aligned}
&S \to a \\
&S \to b \\
&S \to c \\
&S \to (S + S) \\
&S \to (S \cdot S) \\
&S \to (S)^*
\end{aligned}
$$

Exercise 6.20. Find the language of this grammar

$$
\begin{aligned}
&S \to S\ S \\
&S \to l\ S\ r
\end{aligned}
$$

(l is for *left*, and r is for *right*.)

7

Semantics I: Compositionality and Sentential Logic

In this chapter we consider first some of the goals of a semantic analysis for a language and we illustrate a semantic analysis for a particularly simple language, that of Sentential Logic (SL). Then we enrich our analysis to include a language with some (not all) of the linguistic complexity studied in Chapter 4.

7.1 Compositionality and Natural Language Semantics

7.1.1 Goals of semantic analysis

Compositionality. Our primary way of understanding a complex novel expression is understanding what the lexical items it is composed of mean and how expressions built in that way take their meaning as a function of the meanings of the expressions they are built from (beginning with the lexical items). We illustrate this conceptually with our semantics for Sentential Logic.

Semantic characterization of syntactic phenomena. In practice syntactic and semantic analysis are partially independent and partially dependent. So a variety of cases arise where the judgments that an expression is grammatical seem to be decided on semantic grounds (See chapters 8, 10 and 11). Here are two examples. First, negative elements like *not* and *n't* license the presence of *negative polarity items* (npi's) within the P_1 they negate:

(1) a. Sue hasn't <u>ever</u> been to Pinsk.
b. *Sue has <u>ever</u> been to Pinsk.

However some subject DPs also license npi's, as in (2a) but not (2b):

(2) a. No student here has <u>ever</u> been to Pinsk.

b. *Some student here has ever been to Pinsk.

These judgments give rise to two linguistic problems: (1) How to define the class of subject DPs which, like *no student*, license npi's in their P_1. This class must be defined in order to define a grammar for English. And (2) what, if anything, do these DPs have in common with *not* and *n't*? The best answer we have to date is stated in semantic terms, specifically in terms of the denotations of the Dets used in the subject DP.

Second, a long standing problem in generative grammar is the characterization of the DPs which occur naturally in Existential-There contexts:

(3) a. Are there *more than two students* in the class?
b. *Are there *most students* in the class?

Again the best answers that linguists have found are semantic in nature: they are those DPs built from Dets whose denotations satisfy a certain condition.

Issues of expressive power. Given an adequate semantic analysis of a class of expressions in natural language we can study that analysis to uncover new purely semantic regularities about the language. For example, we can show that natural languages present quantifier expressions which are not definable in first order logic (Chapter 9), and we can show that Det denotations quite generally satisfy a logically and empirically non-trivial condition known as Conservativity (Chapter 11).

7.1.2 Semantic Facts

Crucial to each of the three goals above is that we have a clear sense of the facts that a semantic analysis of natural language must account for[1]. That is, we need a way of evaluating whether a proposed semantic analysis is adequate or not. The facts we rely on are the judgments by competent speakers that a given expression has, or fails to have, a certain semantic property. More generally a semantic analysis of a language must explicitly predict that two (or more) expressions stand in a certain semantic relation if and only if competent speakers judge that they do. Pre-theoretically to say that a property P of expressions

[1]This problem arises in syntactic study as well, but it is less pressing as we begin with a large range of syntactic facts our theory should predict. Namely facts of the form *Every cat chased every dog* is an expression of English whereas *Cat dog every every chased* is not. As we pursue syntactic analysis more deeply than was done in Chs 3 and 4 further non-trivial methodological issues arise, but we do not pursue them in this book.

is *semantic* is just to say that competent speakers decide whether an expression has P or not based on the *meaning* of P. Similarly a relation between expressions is semantic just in case whether expressions stand in that relation depends on the meaning of the expressions. The best understood semantic relation in this sense is *entailment*, introduced briefly in Chapter 4. We repeat and expand that definition:

Definition 7.1. A sentence P *entails* ($\models$) a sentence Q iff Q is true in every situation (model) in which P is true. More generally a set K of sentences *entails* a sentence Q iff Q is true in every situation (model) in which the sentences in K are simultaneously true.

In Chapter 4 we gave examples using manner adverbs:

(4) a. John walked rapidly to the post office. $\models$ John walked to the post office.
b. Sue smiled mischievously at Peter. $\models$ Sue smiled at Peter.

In a situation in which *Sue smiled mischievously at Peter* is true then, obviously, in that situation, it is true that *Sue smiled at Peter*. Our judgments of entailment here are good, even though we may be unclear about precisely what a smile must be like to be mischievous. But the judgment of entailment doesn't require that we know precisely the truth conditions of the first sentence, it just requires an assessment of *relative* truth conditions. *If* a situation suffices to make the first true, does it also suffice to make the second true?

Thus one adequacy condition on a semantic analysis of English is that it predict the entailments in (4). More generally, an adequate semantic analysis of English must tell us that an English sentence P entails an English sentence Q if and only if competent speakers judge that it does. Thus an adequate semantic analysis must correctly predict the judgments of entailment and non-entailment by competent speakers.

Our observations here incorporate an important assumption concerning the nature of truth, one of our fundamental semantic primitives. Namely we treat truth as a relation between a sentence in a language and "the world" or "the situation we are talking about", notions we shortly represent more formally as "models". The truth value—True or False—of a given sentence may vary according to how the world is. A simple sentence such as *Some woodworker likes mahogany* is true in some situations and false in others. It depends on what woodworkers there are in the situation and what they like. This is why our definition of entailment quantifies over situations (models). It says that for P to entail Q it must be so that in *each* situation in which P is true Q is true.

7.1.3 Further Adequacy Criteria for Semantic Analysis

Semantic ambiguity. Not uncommonly an expression is felt to express two or more distinct meanings. In such a case obviously all the meanings must be represented. To take a classical example, the sentence *Flying planes can be dangerous* is semantically ambiguous. On the one hand the subject phrase *flying planes* can refer to the act of flying planes, and so is presumably dangerous to those who fly them. On this interpretation the subject phrase is grammatically singular, as is evident in the choice of singular *is* in *Flying planes is dangerous.* But the original sentence has another interpretation on which it means that planes that are flying are dangerous, presumably to those in their vicinity. In this case the subject phrase is plural, as seen in *Flying planes are dangerous.* In our original example, *Flying planes can be dangerous*, the Predicate Phrase is built with a modal *can.* (Some other modals, of which there are about 10 in English, are *might*, *may*, *must*, *should*, *will*, *would* and *could*). Modals in English neutralize verb agreement. One says equally well *Johnny can read* and *the children can read* with no change in the form of the predicate despite the first having a singular subject and the second a plural one.

Exercise 7.1. Each of the expressions below is semantically ambiguous. In each case describe the ambiguity informally.

a. The chickens are ready to eat.
b. France fears America more than Russia.
c. John thinks he's clever and so does Bill.
d. Ma's home cooking.
e. John and Mary or Sue came to the lecture.
f. John didn't leave the party early because the children were crying.

Of the expressions in our model language $\mathcal{L}$(Eng) two cases of possible ambiguity have arisen. First, (Kim smiled, S) was *syntactically* ambiguous according as *Kim* had category NP or S/(NP\S). But this S is not felt to be semantically ambiguous. So a semantic interpretation of $\mathcal{L}$(Eng) must show that the two syntactic analyses are compositionally interpreted to yield the same result. And second, recall DP scope ambiguities in Ss like *Some student praised every teacher.* In this chapter we represent the object narrow scope reading. The object wide scope reading, *Every teacher has the property that some student praised him* is treated in Chapter 9. A related type of ambiguity is the transparency/opacity (= *de re / de dicto*) one in Ss like (5):

(5) Sue thinks that the man who won the race was Greek.

On the opaque (*de dicto*) reading of *the man who won the race* we understand that Sue thinks that the winner was Greek. Sue may have no direct knowledge of who the winner was, she may just know that all the contestants were Greek men so obviously the winner was Greek. On the transparent reading of this DP, (5) is interpreted like *The man who won the race has the property that Sue thinks that he was Greek.* Here Sue has an opinion about a certain individual, namely that that individual is Greek, but she may not even know that he won the race.

Variations on this type of ambiguity are rife in the analysis of expressions involving Sentence Complements of verbs of thinking and saying, especially in the philosophical literature where such verbs are said to express *propositional attitudes*. It is among the reasons we do not attempt a quick semantic analysis here. These problems have no fully agreed upon solution in the literature.

Selection restrictions. For most of the expression types considered in $\mathcal{L}$(Eng) we find that choices of the slash category expression semantically constrain the choice of expression in the denominator category. Here are some examples.

Adjective + Noun. It makes sense to speak of a skillful or accomplished writer, but not of a skillful or accomplished faucet. Faucets are not the kinds of things that can be skillful or accomplished—those adjectives require that the item modified denote something animate at least. We say that adjectives *select* (impose *selection restrictions* on) their N arguments.

Predicate modifiers. These exhibit similar selection properties as adjectives. We use # to indicate a selection restriction violation and a check ✓ for selection restriction satisfaction.

(6) a. He solved the problem ✓ in an hour / # for an hour.
b. He knocked at the door # in an hour / ✓ for an hour.

Thus a repetitive or durative action can be modified by durational phrases such as *for an hour* but not by modifiers like *in an hour*. In contrast an accomplishment or achievement predicate like *solve the problem*, which is over in an instant when it is over, can sensibly take modifiers like *in an hour* but not duratives like *for an hour* (See Dowty (1982)).

Determiners + Noun. Dets also place some selection requirements on the Ns they determine. *Many students* is natural, #*Many gold* is senseless. In contrast *Much gold* is sensible and #*Much students* is not. Ns like *gold*, *butter*, *hydrogen* are called *mass nouns*, whereas ones like *student*, *brick*, and *number* are called *count nouns*. And Dets may at least select for mass or count. Many abstract nouns, like *honesty*, *sincerity*,and *honor* behave like mass nouns in this respect.

Predicates + Argument. P_1s impose selection restrictions on their subjects: *The witness lied* is fine, but #*The ceiling lied* is bizarre since ceilings aren't the kind of thing that can lie—they are too low on the chain-of-being hierarchy. Also P_2s impose selection restrictions on their object arguments. It makes sense to say that *John peeled an orange* or *a grape*, but not that *he peeled a puddle* or *a rainstorm*.

Beyond pointing out their existence we do not study selection restrictions in this text. Our examples are in general chosen to satisfy selection restrictions.

Sense dependency. Sense dependency is a phenomenon inversely related to selection restrictions whereby the interpretation of the slash category expression is conditioned by the denotation of the denominator category expression. Consider again Adjective + Noun constructions. When we speak of a flat road or table top we interpret *flat* to mean "level, without bumps or depressions". But when we speak of flat beer or champagne we mean "having lost its effervescence". And a flat tire is one that is deflated, a flat voice is one that is off-key. So the precise interpretation of *flat* is conditioned by its argument.

Predicates also have their interpretation conditioned by the nature of their arguments. In *cut your finger*, *cut* means to make an incision in the surface of. But in *cut the roast* or *the cake*, *cut* means to divide into portions for purposes of serving. In *cut prices* or *working hours*, *cut* means to reduce along a continuously varying dimension.

Sense dependency is not one of the well studied semantic relations in the linguistic literature, but dictionaries note them. The examples here are taken from Keenan (1979).

Presupposition. Presupposition is a well studied relation, one which plays an important role in many semantic and pragmatic studies. Informally we say that a sentence P (*logically*) *presupposes* a sentence Q iff Q is an entailment of P which is preserved under Yes-No questioning and "natural" negation. Consider for example the classical (7a).

(7) a. The king of France is bald.
b. France has a king.
c. Is the king of France bald?
d. The king of France isn't bald.

Clearly (7a) entails (7b)—if the king of France is bald then France must indeed have a king. And that information is not questioned in

(7c) or denied in (7d), hence (7b) is a presupposition of (7a).

Presupposition can be used to distinguish meanings of predicates. Consider first:

(8) a. It is true that Fred took the painting.
b. Fred took the painting.
c. Is it true that Fred took the painting?
d. It isn't true that Fred took the painting.

Though (8a) entails (8b), (8b) is not presupposed by (8a). The information in 8b) is questioned in (8c). Someone who asks (8c) is asking whether the embedded S, (8b), is true or not. And similarly (8d) does deny the information in (8b). We can replace *true* with *false* or *probable* and argue, even more easily, that they do not presuppose the b-sentence either. By contrast consider (9a).

(9) a. It is strange that Fred took the painting.
b. Fred took the painting.
c. Is it strange that Fred took the painting?
d. It isn't strange that Fred took the painting.

Here (9a) does seem to entail (9b). And in (9c) we are not asking whether Fred took the painting, we are accepting that and asking whether that fact is strange or not. Similarly in (9d) we are just denying the strangeness of the fact, but not the fact itself. It seems then that (9a) does presuppose (9b). Moreover *strange* can be replaced by dozens of other presuppositional adjectives: *amazing*, *unsurprising*, *pleasing*, *ironic*, etc.

There is then a systematic difference between the predicates in (8) and those in (9), one that is revealed by observing that they behave differently with regard to whether the embedded S is presupposed or not.

In general presupposition is a relation that is most useful is discerning how information is packaged in a sentence, as opposed to the absolute quantity of information. To see this compare (10a) and (10b):

(10) a. John is the one doctor who signed the petition.
b. John is the only doctor who signed the petition.
c. Exactly one doctor signed the petition.

Each of (10a) and (10b) entails the other, which means that they are true in the same situations and so in that sense they express the same absolute information. They both entail (10c) for example. But (10a) and (10b) present their information somewhat differently. Compare their natural negations:

(11) a. John isn't the one doctor who signed the petition.
b. John isn't the only doctor who signed the petition.

Now (11a) still entails (10c), it only denies that John is that doctor. But (11b) denies that John was the only doctor who signed, implying thereby that there was an additional doctor who signed. So (11b) does not entail (10c). And it seems then that (10a) and (10b), while logically equivalent, differ in that (10a) presupposes (10c) whereas (10b) does not. Questioning or denying (10b) does not preserve that information.

Now, as we have seen, Ss in natural language are syntactically complex objects, and there are normally infinitely many of them. So we cannot just list the set of Sentence interpretations in English. Rather we must show for each S how it's interpretation is constructed from the interpretation of the lexical items which occur in it. Those we can list—that is what *dictionaries* do—since the number of lexical items in a natural language is finite. Otherwise the recursive construction of interpretations follows the same steps as the recursive construction of the expressions themselves. And in Eng there are only two rule sets that build complex expressions from simpler ones: the rules of Function Application and the Coordination Rule. A more complete grammar for English would almost certainly have more rules.

7.1.4 A basic example of compositionality

Our grammar Eng in Chapter 4 allows us to form coordinations of Ss using *and* and *or* (and with some modification, *neither...nor...*). Such syntactically complex Ss are called *boolean compounds* of the ones they are built from. *And*, *or*, *neither...nor...* and *not* are called *boolean connectives*.

Now in a given situation, the truth value of a boolean compound of Ss is uniquely determined by—is a function of— the truth values in that situation of the Ss it combined. That is, the boolean connectives are *truth functional*. In a situation in which P is true and Q is true we infer *Both* P *and* Q is true, *Either* P *or* Q is true, and *Neither* P *nor* Q is false. Many subordinate conjunctions that build an S from two others are not truth functional. Imagine a situation in which *John left the party early* and *The children were crying* are both true. The sentence *John left the party early because the children were crying* may still be either true or false. So its truth value is not determined by the truth of its component sentences. Thus *because* is not a truth functional connective. *Sentential Logic* is used by logicians and philosophers to study the meanings of boolean (truth functional) connectives. Below we present it explicitly as it illustrates in a simple form how we may define a language and compositionally interpret it. And as a result it provides a convenient vehicle for studying basic issues in both syntax and semantics.

7.2 Sentential Logic

The language SL of *Sentential Logic* (also called *Propositional Logic*) consists of denumerably many *atomic formulas* $P1$, $P2$,... closed under combinations with *and*, *or* and *not*. So for example (($P1$ and not $P5$) or ($P5$ and not $P1$)) is an element of SL, whose members in general are called *formulas*. Here is a more precise definition. Recall that for V a set, V^* is the set of finite sequences of elements of V.

7.2.1 The Syntax of SL

Definition 7.2. $\text{AF} =_{df} \{\langle \text{'}P\text{'}, n\rangle \mid n \in \mathbb{N}\}$.

We usually write simply P_n for $\langle \text{'}P\text{'}, n\rangle$. By '$P$' we just mean the letter P. Elements of AF are called *atomic formulas.*

Definition 7.3. $\text{V} =_{df} \text{AF} \cup \{\text{and}, \text{or}, \text{not},), (\}$

We define one unary function NEG on V^* and two binary functions AND, OR on V^* as follows (writing simply uv for $u \frown v$).

Definition 7.4. $\text{NEG}(\varphi) = \text{not } \varphi, \text{AND}(\varphi, \psi) = (\varphi \text{ and } \psi)$, $\text{OR}(\varphi, \psi) = (\varphi \text{ or } \psi)$.

Definition 7.5. SL is the closure of AF under NEG, AND, and OR.

Thus SL is the set of strings that can be built starting with the atomic formulas and applying the NEG, AND, and OR functions any finite number of times. An explicit definition of this closure, on the model of our earlier one for Cat_{Eng}, is given below. Here (and elsewhere) we write $x, y \in A$ as a shorthand for "$x \in A$ and $y \in A$".

(12)
  i. Set $\text{SL}_0 = \text{AF}$, and
  ii. for all natural numbers n,
  $\text{SL}_{n+1} = \text{SL}_n \cup \{\text{NEG}(\varphi) \mid \varphi \in \text{SL}_n\}$
  $\cup\{\text{AND}(\varphi, \psi) \mid \varphi, \psi \in \text{SL}_n\} \cup \{\text{OR}(\varphi, \psi) \mid \varphi, \psi \in \text{SL}_n\}$.
  iii. Then $\text{SL} =_{df} \{\tau \in \text{V}^* \mid \text{for some } n, \tau \in \text{SL}_n\}$.
  So $\text{SL} =_{df} \bigcup_{n \in \mathbb{N}} \text{SL}_n$.

Then SL provably has the following three basic properties:

Theorem 7.1.

a. *$AF \subseteq SL$.*

b. *SL is closed under NEG, AND, OR. That is, if $\varphi, \psi \in SL$ then $NEG(\varphi), AND(\varphi, \psi)$ and $OR(\varphi, \psi)$ are in SL as well.*

c. *If a set K includes all the atomic formulas and is closed under NEG, AND, and OR then $SL \subseteq K$. (This is what is meant by saying that SL is the* least *subset of V^* which includes the atomic formulas and is closed under NEG, AND, and OR).*

Some abbreviations.

i. We often write & instead of *and*.

ii. Ss of the form $(\varphi \;\&\; \psi)$ are called *conjunctions*; φ and ψ are its *conjuncts*.
Ss of the form (φ or ψ) are called *disjunctions*; φ and ψ are its *disjuncts*.

iii. For $\varphi, \psi \in$ SL we use $(\varphi \to \psi)$ to abbreviate ((not φ) or ψ). $(\varphi \to \psi)$ is called a *conditional* formula; φ called its *antecedent* and ψ its *consequent*.

iv. Similarly we write $(\varphi \leftrightarrow \psi)$ to abbreviate $((\varphi \to \psi) \;\&\; (\psi \to \varphi))$. Formulas of the form $(\varphi \leftrightarrow \psi)$ are called *biconditionals*

.

In what follows it will be useful to be able to refer to the atomic formulas which occur one or more times in a given formula φ. Here is an explicit recursive definition, one that illustrates the recursive format:

Definition 7.6. For all $\varphi, \psi \in$ SL,

a. $\text{AF}(Pn) = \{Pn\}$ for all atomic formulas Pn,
b. $\text{AF}(\text{NEG}(\varphi)) = \text{AF}(\varphi)$,
b. $\text{AF}(\text{AND}(\varphi, \psi)) = \text{AF}(\varphi) \cup AF(\psi)$, and
c. $\text{AF}(\text{OR}(\varphi, \psi)) = \text{AF}(\varphi) \cup AF(\psi)$.

AF(φ) is the set of atomic formulas which occur in φ.

AF here is a recursively defined function from SL into $\mathcal{P}$(SL), the power set of SL, since it associates with each formula of SL a set of formulas. That AF is *well defined* depends on the fact that NEG, AND, and OR are *unambiguous*. Imagine for example that there was a formula σ such that $\sigma = NEG(\tau)$ for some $\tau \in$ SL and also that $\sigma = \text{OR}(\varphi, \psi)$ for some $\varphi, \psi \in$ SL. Then Definition 7.6 would say that $\text{AF}(\sigma) = \text{AF}(\tau)$ and also that $\text{AF}(\sigma) = \text{AF}(\varphi) \cup \text{AF}(\psi)$. Likely these two sets are not the same, so AF would not be a function, it would associate two different values with σ. But in fact this situation provably does not arise:

Theorem 7.2. *SL is syntactically unambiguous. That is, (a)–(c) below hold:*

a. *Each generating function NEG, AND, OR is one to one,*
b. *The ranges of any two of NEG, AND, OR are disjoint,and*

c. AF and the range of any of NEG, AND, and OR are disjoint.

Thus no formula gets into SL in more than one way. Here is a stepwise computation using Definition 7.6:

(13) AF((not $P2$ or $P3$)&($P3$ or $P4$))
$= \text{AF}(\text{not } P2 \text{ or } P3) \cup \text{AF}(P3 \text{ or } P4)$
$= \text{AF}(\text{not } P2) \cup \text{AF}(P3) \cup \text{AF}(P3) \cup \text{AF}(P4)$
$= \text{AF}(P2) \cup \{P3\} \cup \{P3\} \cup \{P4\}$
$= \{P2\} \cup \{P3, P4\}$
$= \{P2, P3, P4\}$

Exercise 7.2. Compute stepwise each of the following:

a. AF($P5$ or not $P5$)
b. AF(not ($P5$ or $P6$))
c. AF($P6 \leftrightarrow P6$)
d. AF(($P1 \rightarrow P9$) or ($P1 \rightarrow$ not $P9$))

(Replace the defined formulas here with the ones that define them.)

7.2.2 The semantics of SL

To simplify notation in what follows consider the two element set $\{T, F\}$ whose elements we call *truth values*, with T = true, and F = false. This set is commonly noted $\{0, 1\}$ in the literature, with 0 = false and 1 = true, but here we stick with $\{T, F\}$ for mnemonic reasons. We define one unary function called *complement* noted $\neg$ on $\{T, F\}$ and two binary functions, *meet* $\wedge$ and *join* $\vee$ by giving their tables:

(14) a.

X	$\neg X$
T	F
F	T

b.

X	Y	$X \wedge Y$	$X \vee Y$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

If a formula φ in our logic has truth value X then its negation has truth value $\neg X$. And if formulas φ, ψ have truth values X and Y respectively then their conjunction has truth value $(X \wedge Y)$ and their disjunction has truth value $(X \vee Y)$. More formally we define:

Definition 7.7. A *model* for SL is a function $v : \text{AF} \rightarrow \{T, F\}$. v is often called a *valuation*.

Definition 7.8. For each model v we define a function v^* from SL into $\{T, F\}$ by setting:

a. $v^*(Pn) = v(Pn)$, for all atomic formulas Pn,
b. $v^*(\text{NEG}\varphi) = \neg(v^*(\varphi))$,
c. $v^*(\text{AND}(\varphi, \psi)) = v^*(\varphi) \wedge v^*(\psi)$, and
d. $v^*(\text{OR}(\varphi, \psi)) = v^*(\varphi) \vee v^*(\psi)$.

v^* is called an *interpretation* of SL.

We note without argument that for each v, v^* is a function, one which extends v. That is, $Dom(v) \subseteq Dom(v^*)$ and for all $\varphi \in Dom(v)$, $v^*(\varphi) = v(\varphi)$. Further, each function from SL into $\{T, F\}$ which satisfies conditions (b)–(d) extends some valuation v, that is, it is an interpretation of SL.

The informal idea behind this semantics is that the atomic formulas represent *independent* claims we can make about the world. If we are in a model in which $P5$ is true we have no predictability of the truth of any other atomic formula. $P7$ for example might be true or it might be false. That is, there are models v in which $v(P5) = T$ and $v(P7) = T$ and other models v' in which $v'(P5) = T$ and $v'(P7) = F$. However once we are given a model v—so we know the truth value of each atomic formula, then the truth values of the syntactically complex formulas are uniquely determined by the stipulations in (b)–(d).

We emphasize that our definition of interpretation is fully *compositional*. The interpretation (truth value) of a complex formula is uniquely determined by the interpretations (truth values) of the formulas it is built from.

A seemingly obvious property of SL is that the truth of a formula depends *only* on the truth of the atomic formulas which occur in it. Formally we have:

Theorem 7.3 (The Coincidence Lemma)**.** *For all $\varphi \in SL$ and all models v and u, if $v(Pn) = u(Pn)$ for all atomic formulas occurring in φ then $v^*(\varphi) = u^*(\varphi)$.*

Proof. By induction on formula complexity. The main idea is: we let K be the set of formulas in SL for which the theorem holds. Then we show that K contains all the atomic formulas and is closed under the functions AND, OR, and NEG. Then by Theorem 7.1c, SL $\subseteq K$, proving the theorem.

More explicitly now, set

$$K = \{\varphi \in \text{SL} \mid \text{for all models } u, v \text{ if } u(P) = v(P) \\ \text{for all } P \in \text{AF}(\varphi) \text{ then } u^*(\varphi) = v^*(\varphi)\}.$$

Step 1: All atomic formulas are in K. Let Pm be an arbitrary atomic formula. Then $\mathrm{AF}(Pm) = \{Pm\}$, so if u, v are models that assign the same value to the elements of $\mathrm{AF}(Pm)$ then, trivially, they assign the same value to Pm.

Step 2: Show that K is closed under NEG. Let $\varphi \in K$, show that $\mathrm{NEG}(\varphi) \in K$. Let u, v be arbitrary models, assume they assign the same values to Pn in $\mathrm{AF}(\mathrm{NEG}(\varphi))$. We must show that

$$u(\mathrm{NEG}(\varphi)) = v(\mathrm{NEG}(\varphi)).$$

But $\mathrm{AF}(\mathrm{NEG}(\varphi)) = \mathrm{AF}(\varphi)$. Hence, by the induction hypothesis, $u(\varphi) = v(\varphi)$. Therefore

$$u(\mathrm{NEG}(\varphi)) = \neg u(\varphi) = \neg v(\varphi) = v(\mathrm{NEG}(\varphi)).$$

Step 3: Show that K is closed under AND. Let $\varphi, \psi \in K$. Show that $\mathrm{AND}(\varphi, \psi) \in K$. Now $\mathrm{AF}(\mathrm{AND}(\varphi, \psi)) = \mathrm{AF}(\varphi) \cup \mathrm{AF}(\psi)$. So if u and v agree on all the Pn in $\mathrm{AF}(\mathrm{AND}(\varphi, \psi))$ then they agree on the Pn in $\mathrm{AF}(\varphi$ and on the Pn in $\mathrm{AF}(\psi)$. Thus by the induction hypothesis $u(\varphi) = v(\varphi)$ and $u(\psi) = v(\psi)$. So

$$u(\mathrm{AND}(\varphi, \psi)) = u(\varphi) \wedge u(\psi) = v(\varphi) \wedge v(\psi) = v(\mathrm{AND}(\varphi, \psi)).$$

Step 4: Show analogously to Step 3 that K is closed under OR. Thus, since K contains all the atomic formulas and is closed under NEG, AND, and OR, we infer that $\mathrm{SL} \subseteq K$, proving the theorem. □

We turn now to the definition of *entailment* and related notions on SL.

Definition 7.9.

a. For $\varphi, \psi \in \mathrm{SL}$, $\varphi \models \psi$ (read: φ *entails* ψ) iff for all models v for SL, if $v^*(\varphi) = T$ then $v^*(\psi) = T$.
   $\varphi \models \psi$ is often read "φ logically implies ψ".
b. For all $K \subseteq \mathrm{SL}$, all $\varphi \in \mathrm{SL}$, $K \models \varphi$ iff for all models v,
   if $v^*(\tau) = T$ for all $\tau \in K$ then $v^*(\varphi) = T$.

Note that Definition 7.9a is just the special case of Definition 7.9b where K is the unit set $\{\varphi\}$. We normally write $\varphi \models \psi$ rather than $\{\varphi\} \models \psi$.

Definition 7.10. For $\varphi \in \mathrm{SL}$, φ is *logically true* (*valid*, a *tautology*) iff for all models v, $v^*(\varphi) = T$. To say that φ is logically true we write $\models \varphi$.

Theorem 7.4. *For $\varphi \in SL$, φ is logically true iff $\emptyset \models \varphi$.*

The significance of Theorem 7.4 is that logical truth (validity) is not a special case of entailment.

Definition 7.11. For $\varphi, \psi \in$ SL, φ is *logically equivalent* to ψ, noted $\varphi \equiv \psi$, iff for all models v, $v^*(\varphi) = v^*(\psi)$.

Theorem 7.5. *For $\varphi, \psi \in SL$,*

a. $\varphi \equiv \psi$ *iff* $\varphi \models \psi$ *and* $\psi \models \varphi$,
b. $\varphi \equiv \psi$ *iff* $\models (\varphi \leftrightarrow \psi)$,
c. $\varphi \models \psi$ *iff* $\models (\varphi \rightarrow \psi)$.

From Theorem 7.5c we carefully distinguish $\models$ from $\rightarrow$. A conditional formula $(\varphi \rightarrow \psi)$ may be true in some interpretations and false in others. But $\varphi \models \psi$ is simply true or false. For example the formula $(P1 \rightarrow P3)$ is true in a model v in which $v(P1) = F$ or $v(P1) = v(P3) = T$. But it is false in models v' in which $v'(P1) = T$ and $v'(P3) = F$. In contrast $P1 \models P3$ is simply false, since it is not the case that $P3$ is interpreted as T in every model in which $P1$ is interpreted as T. The models v' just mentioned are examples.

Note that to prove a formula of the form $(\varphi \rightarrow \psi)$ in mathematical discourse it suffices to consider the case when φ is true and then prove that ψ is true. If φ is false then $(\varphi \rightarrow \psi)$ is true no matter what truth value ψ has.

Equivalence Relations. In Definition 7.11 we defined a semantic relation, *logical equivalence*, noted $\equiv$, on the formulas of Sentential Logic. This relation is representative of a class of widely used relations in mathematical discourse called *equivalence relations*. Roughly, an equivalence relation R defined on a set A relates objects x and y in A if they are identical in some respect or another. In the limit they may be identical in every respect, that is, they are the exact same object. Indeed, absolute equality $=$ is an equivalence relation. But the utility of equivalence relations lies in the fact that we can ignore certain differences between objects, as we did with $\equiv$ above. If we are just interested in studying the entailments (logical consequences) that a formula may have then there is no need to distinguish between (not P & not Q) on the one hand and not(P or Q) on the other, since they have the same entailments as they are logically equivalent (always have the same truth value). Formally we define a relation to be an equivalence relation as follows.

Definition 7.12. A binary relation R on a set A is an *equivalence relation* iff:

a. R is reflexive (for all $x \in A$, xRx),
b. R is symmetric (for all $x, y \in A$, $xRy \Rightarrow yRx$), and
c. R is transitive (for all $x, y, z \in A$, $(xRy \;\&\; yRz) \Rightarrow xRz$).

Exercise 7.3.

a. State explicitly the three conditions that must be met for $\equiv$ defined above to be an equivalence relation.
b. We define the equi-cardinality relation $\approx$ on $\mathcal{P}(\mathbb{N})$ by: $A \approx B$ iff there is a bijection from A to B (that is, A and B have the same cardinality). State explicitly the three things you must show to prove that $\approx$ is an equivalence relation. For each of these three say what justifies its truth.
c. Consider the relation $\cong$ defined on the set of simple trees with nodes drawn from $\mathbb{N}$. Is $\cong$ an equivalence relation? If so, say why it is reflexive, symmetric, and transitive.
d. Which property of equivalence relations makes them distinct from partial order relations (such as $\leq$ in arithmetic, or $\subseteq$ on sets, or *dominates* on nodes of a tree)?

Given an equivalence relation R on a set A, we may group equivalent objects together and, often, just study the behavior of these *equivalence classes.*

Definition 7.13. Given an equivalence relation R and an object $x \in A$, we set $[x]_R = \{b \in A \mid aRb\}$. $[x]_R$ is called the *equivalence class of x modulo R.*

When R is clear from context we simply write $[x]$. $[x]$ is a subset of A but it is never $\emptyset$. Why not?

Theorem 7.6. *For all $x, y \in A$, $[x] = [y]$ iff xRy.*

Proof. ($\Longrightarrow$) Let $[x] = [y]$. We must show that xRy. Since $x \in [x]$ because xRx and $[x] = [y]$ we have that $x \in [y]$. So by the definition of $[y]$ it follows that yRx. Thus, because R is symmetric we conclude that xRy
($\Longleftarrow$) Suppose xRy. We must show that $[x] = [y]$. Let z be arbitrary in $[x]$. Then xRz, so zRx, whence by the transitivity of R we have that zRy. Since R is symmetric it follows that yRz, and so $z \in [y]$. Since z was arbitrary, $[x] \subseteq [y]$. By analogous reasoning, $[y] \subseteq [x]$, so equality holds. $\square$

Corollary 7.7. *For R an equivalence relation on A, $\{[x] \mid x \in A\}$ is a partition of A.*

A *partition* of a set A is a collection of non-empty subsets of A such that each element x in A is in exactly one of the subsets. So any two of the subsets are disjoint. The corollary follows immediately from Theorem 7.6 given that $[x]$ is never empty.

Returning to Sentential Logic again, observe first that the entailment relation $\models$ is reflexive since any formula entails itself. This relation is also transitive, but it is not antisymmetric, since for any formula φ, $\varphi \models$ not not φ and conversely, but φ and not not φ are not the same formula. But consider now what happens when we take the corresponding relation on the equivalence classes of formulas.

Definition 7.14. For all formulas φ, ψ in Sentential Logic, $[\varphi] \leq [\psi]$ iff $\varphi \models \psi$.

We purport here to define the relation $\leq$ on the set of equivalence classes of formulas (under the logical equivalence relation) in terms of the relation $\models$ on the formulas themselves. We must be careful in such cases. Suppose for example we could find some $\varphi' \in [\varphi]$ such that φ' failed to entail some ψ even though φ did entail ψ. Then we would be claiming that $[\varphi] \leq [\psi]$ and $[\varphi] \not\leq [\psi]$, which is to say that the $\leq$ relation would not be well-defined. Fortunately this cannot happen, since $\varphi \models \psi$ means that every valuation that makes φ true also makes ψ true. And the valuations that make φ true are exactly those that make φ' true since they are logically equivalent. So in defining the $\leq$ relation as we did, we see that whether is holds or not does not depend on the choice of representative we pick from $[\varphi]$ (or from $[\psi]$).

Observe now that the $\leq$ relation is a partial order; in particular it is antisymmetric since if $[\varphi] \leq [\psi]$ then $\varphi \models \psi$. And if $[\psi] \leq [\varphi]$ then $\psi \models \varphi$, so φ and ψ are logically equivalent (if one could be true under some valuation in which the other was false the one would fail to entail the other). So once we have traded in the entailment relation on formulas for the $\leq$ relations on the equivalence classes of formulas we are then working with a familiar ordering relation. More about this order can be found in Chapter 8.

Decidability. Sentential Logic has the pleasing property that there is a general mechanical procedure (an *algorithm*) for deciding whether an arbitrary formula φ is logically true or not. The procedure is called a *Decision Procedure* and SL is said to be *decidable*. Similarly there is a procedure for deciding whether arbitrary formulas φ, ψ are logically equivalent, or whether one entails the other. The procedures all use truth tables, illustrated below.

Consider the formula $((P2$ & $P5)$ or (not $P2$ & not$P5$)). To test whether it is logically true we must evaluate its truth under all interpretations. But by the Coincidence Lemma we need only consider a model v in so far as it assigns truth values to $P2$ and $P5$, the atomic formulas occurring in it. Any two interpretations which assign the same values to $P2$ and to $P5$ must assign the same value to the formula

(($P2$ & $P5$) or (not $P2$ & not $P5$)). Now there are just two ways we can assign truth values to $P2$, and for each of those there are two ways to assign truth values to $P5$. So there are a total of 4 combinations of truth values that $P2$ and $P5$ can take jointly. So let us list all cases, writing under each atomic formula the value we assign it and then computing the truth value of the entire formula by writing the truth value of a derived formula under the connective (&, or, not) used to build it. Here is what this procedure yields in this case:

(15)	$P2$	$P5$	(($P2$	&	$P5$)	or	(not$P2$	&	not$P5$))
	T	T		T		T	F	F	F
	T	F		F		F	F	F	T
	F	T		F		F	T	F	F
	F	F		T		F	T	T	T

The first line in this truth table says that if $P2$ and $P5$ are both true then the entire formula is a disjunction of a true formula with a false one, and thus is true. The second line says that if $P2$ is interpreted as true and $P5$ as false then the entire formula is a disjunction of a false formula with a false one and thus is false. Lines 3 and 4 are computed similarly. Thus (15) is not logically true as it is not true under all assignments of truth values to the atomic formulas occurring in it. To show that it is not logically true it suffices to exhibit line 2 or line 3 in the truth table.

Generalizing, since any formula in SL is built from just finitely many atomic formulas (Theorem 7.5) we can decide the validity of any such formula by constructing a truth table.

Similarly to show that two formulas are not logically equivalent it suffices to illustrate one line of their truth table in which they have different values. If they have the same value for all lines then they are logically equivalent. And finally to show that some φ entails some ψ you must show that for each line of the truth table for φ which makes it true, ψ is also true in that case. To falsify the entailment claim it suffices to find one assignment of truth values to the atomic formulas of φ which make φ true but ψ false.

Exercise 7.4. Establish the claims below by exhibiting an assignment of truth values to the atomic formulas in the left hand formula which make it true and the right hand formula false. We write $\nvDash$ for *does not entail*.

a. (not$P4$ or $P7$) $\nvDash$ (not$P7$ or $P4$)
b. (($P4$ or $P7$) & ($P8$ or $P7$)) $\nvDash$ ($P4 or P8$)
c. (($P1$ & $P2$) or $P3$) $\nvDash$ ($P1$ & ($P2$ or $P3$))

Exercise 7.5. For each pair of formulas below show that they are logically equivalent if they are and exhibit a line of their truth table at which they differ if they are not.

a.	i.	$((P \,\&\, Q)$ or $(\text{not}P \,\&\, \text{not}Q))$	ii.	$(P \leftrightarrow Q)$
b.	i.	$(P \to Q)$	ii.	$(\text{not}Q \to \text{not}P)$
c.	i.	$(P \,\&\, Q)$	ii.	$(Q \,\&\, P)$
d.	i.	$(P \to Q)$	ii.	$((P \,\&\, Q) \leftrightarrow P)$
e.	i.	not not$P$	ii.	P
f.	i.	$((P \,\&\, Q)$ or $P)$	ii.	P
g.	i.	$(P \,\&\, (Q$ or $R))$	ii.	$((P \,\&\, Q)$ or $(P \,\&\, R))$

7.2.3 Some reflections on the syntax and semantics of SL

Syntax. We first consider an easily established point which (1) illustrates how to prove claims about SL, and (2) may counteract a confusion students occasionally make.

Theorem 7.8. *For all $\varphi \in SL$, $AF(\varphi)$ is finite.*

(So SL itself is an infinite set, but each element of it is built from just a finite number of atomic formulas).

Proof. Set $K =_{df} \{\varphi \in \text{SL} \mid \text{AF}(\varphi) \text{ is finite}\}$.

(1) We show first that $\text{AF} \subseteq K$. That is, each atomic formula Pn is in K. This is so since $\text{AF}(Pn) = \{Pn\}$ which has just one element and so is finite.

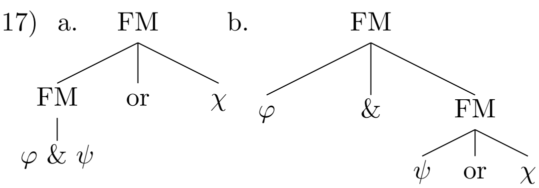
(2) a. K is closed under NEG.
Suppose $\varphi \in K$. We must show that $\text{NEG}(\varphi) \in K$. But $\text{AF}(\text{NEG}(\varphi)) = \text{AF}(\varphi)$ and thus is finite since $\varphi \in K$ so $\text{AF}(\varphi)$ is finite.

b. K is closed under AND and OR.
Let $\varphi, \psi \in K$. Show that $\text{AND}(\varphi, \psi) \in K$. $\text{AF}(\text{AND}(\varphi, \psi)) = \text{AF}(\varphi) \cup \text{AF}(\psi)$, which is finite since the union of two finite sets is finite (if the first has exactly k elements and the second exactly m then the union has at most $k + m$ and so is finite). That $\text{AF}(\text{OR}(\varphi, \psi))$ is finite is shown similarly.

So by Theorem 7.1 $\text{SL} \subseteq K$. And since $K \subseteq SL$ then $K = SL$, that is, each formula in SL contains just finitely many atomic formulas. $\square$

The syntactic role of parentheses: Polish notation. Let φ, ψ and χ be formulas in SL. Then (16a,b) are different formulas, whose derivations may be represented by the trees in (17a,b).

(16) a. $((\varphi \,\&\, \psi)$ or $\chi)$ b. $(\varphi \,\&\, (\psi$ or $\chi))$

(17)


The structures in (17a,b) together with our definition of interpretation make it clear that formulas of these two forms are not logically equivalent. For example in any model in which φ is interpreted as F and χ as T (17a) will be True as it is a disjunction of a true formula with something; in contrast (17b) will be False, as it is a conjunction of a false formula with something.

Now the constituency (derivational history) represented by (16a,b) is coded by the use of parentheses in (15a,b). And (finite) ordered labeled trees satisfying Exclusivity can always be represented this way (subscripting parentheses with labels when necessary). It is natural to wonder if we could simplify the linear representation by eliminating parentheses. The short answer is No. If we omitted all parentheses from (15a,b) then the resulting expressions would be identical and thus would be semantically ambiguous—possibly True, possibly False depending on how we thought the expression was built (and a model). However, a parenthesis-free notation is available if we write the boolean connectives (not, and, or,...) always on the left of the formulas they combine with (called *Polish* notation; *reverse Polish* puts all the connectives after the formulas they combine with). The notation we used is *infix* notation, as the binary connectives occur between their arguments. Here is what the syntax of Polish SL would look like, given less formally than in our original syntax for SL. We use A for *and*, O for *or*, and N for *not* and otherwise assume the abbreviations we gave for the infix notation.

(18) $\mathrm{SL}_{\mathrm{Pol}}$ is the least set such that (a) and (b) below hold:
  a. $\mathrm{AF} = \{P1, P2, \ldots\} \subseteq \mathrm{SL}_{\mathrm{Pol}}$.
  b. If $\varphi, \psi \in \mathrm{SL}_{\mathrm{Pol}}$ then $A\varphi\psi$, $O\varphi\psi$ and $N\varphi \in \mathrm{SL}_{\mathrm{Pol}}$.

Then (16a,b) translate from our infix notation to (19a,b) respectively in Polish:

(19) a. $((\varphi \mathbin{\&} \psi) \text{ or } \chi) =_{\text{trans}} OA\varphi\psi\chi$
  b. $(\varphi \mathbin{\&} (\psi \text{ or } \chi)) =_{\text{trans}} A\varphi O\psi\chi$

Exercise 7.6. Draw the derivation trees for each of the Polish formulas in (19).

Exercise 7.7. Translate each of the formulas below from infix SL into SLPol.

a. $(\varphi \rightarrow \psi)$ b. $((\text{not}\varphi \ \& \ \textit{not} \ \psi) \text{ or } (\varphi \ \& \ \psi))$
c. $((\varphi \ \& \ \text{not}\psi) \text{ or } (\text{not}\varphi \ \& \ \psi))$ d. $(\text{not}\varphi \leftrightarrow \text{not}\psi)$

Exercise 7.8. Translate the formulas below from SL_{Pol} into infix notation.
a. $NA\varphi\psi$ b. $AN\varphi\psi$
c. $AAA\varphi\psi\chi\tau$ d. $OAO\varphi\psi\chi\tau$
e. $O\varphi A\psi O\chi\tau$ f. $A\varphi O\psi A\chi\tau$

Exercise 7.9. Compare the derivation trees for (c) in Exercise 7.8 above with that for your translation into infix notation. Is there any difference between the trees in terms of degree of center embedding as opposed to left or right branching?

We should emphasize here that SL_{Pol} is syntactically unambiguous. That is, the analogue of Theorem 7.2 given earlier holds when we set $\text{AND}(\varphi, \psi) = A\varphi\psi$, $\text{NEG}(\varphi) = N\varphi$, etc. So our recursive definition of interpretation in a model carries over as before.

Naively it seems natural to consider that it is the use of *matching* parentheses in infix notation that accounts for the non-ambiguity of its expressions. Matching parentheses conveniently identify the constituents we interpret in building an interpretation of an entire expression. And one of the (partial) automated checks of syntactic well-formedness is a parity check that rejects a string of symbols if the number of left and right parentheses is different. (They must be the same in any formula of SL since whenever a structure building rule (function) puts in a left parenthesis it also puts in a right one, and conversely.

Exercise 7.10. Show that the set Polish SL sentences is context-free but not regular.

Informed now about parenthesis-free notation fixing the boolean connectives formula initially, we can prove that a syntax like our infix one but which only introduces left parentheses also leads to unambiguous expressions. Here the AND function would be: $\text{AND}(\varphi, \psi) = (\varphi \ \& \ \psi$, and $\text{OR}(\varphi, \psi) = (\varphi \text{ or } \psi$. These expressions look odd, and to our knowledge no one has ever proposed a syntax for SL just like this, but it is easy to do so. In fact, suppose we modified these AND and OR functions so that they introduce their first argument with different shaped parentheses—say round ones for AND and square brackets for OR, thus:

(20) a. $\text{AND}(\varphi, \psi) = (\varphi\psi$ b. $\text{OR}(\varphi, \psi) = [\varphi\psi$

Clearly this grammar is just a notational variant of Polish notation where the (plays the role of A and [the role of O.

Semantics Here we note some standard results concerning semantic properties of SL. In some cases explicit definitions and proofs would recapitulate a significant amount of mathematical logic. It is not our intent to do anything like that. Rather, what the reader should take from section is that given an explicit syntax and interpretation for SL we can in fact make interesting, non-obvious, *linguistic* claims about SL.

Theorem 7.9 (Decidability). *SL is decidable.*

We have already noted that there is an algorithm (mechanical process) which will tell us for any $\varphi \in$ SL that it is true in all models if it is and that it isn't if it isn't. The algorithm essentially says that given any φ write down its truth table and check that each line is T. In a similar way we can mechanically check whether $\varphi \models \psi$ or not (We just check whether (notφ or ψ) is true in all models).

This decidability result should not however go to our heads. To be sure, given any φ it contains only finitely many, say n, atomic formulas so we write them all down, compute all possible combinations of truth values and see if φ is true in each (or that φ is false in one of them). But a moment's reflection tells us that given n atomic formulas, since each one is two valued (either T or F) the number of "possible combinations" above is 2^n. So the number of lines in the truth table quickly gets too large to realistically compute and the problem is said to be *intractable*. For example, for $n = 100$ the number of lines in the truth table, 2^{100}, is a 31 digit number, much greater than the number of microseconds since the Big Bang; see Harel (1987).

Theorem 7.10 (Compactness). *For $S \subseteq SL$ and $\varphi \in SL$, $S \models \varphi$ iff there is a finite subset K of S such that $K \models \varphi$.*

Compactness tells us that in SL the truth of a claim φ cannot depend on infinitely many premises. If φ follows from some infinite set then there is a finite subset from which it follows.

Theorem 7.11 (Interpolation). *For φ, ψ non-trivial (neither is true in all models or false in all models), If $\varphi \models \psi$ then there is a $\tau \in SL$ such that $\varphi \models \tau$ and $\tau \models \psi$ and $AF(\tau) \subseteq AF(\varphi) \cap AF(\psi)$.*

So if a formula non-trivially entails another that fact just depends on the interpretations of the atomic formulas they have in common (See Craig (1957), van Dalen (2004) pg. 48).

We write $S \vdash \varphi$ to say that there is a proof of φ from premises in S. Crucial here is that the notion *proof* is purely syntactic. A proof of φ from premises S is a finite sequence of formulas ending in φ, each of which is drawn from S and marked as a premise or is derived by syntactic rule from earlier formulas in the sequence.

There are just finitely many rules: Here are some candidates: *Conjunction Elimination*: If $(\psi \ \& \ \chi)$ is a line in a proof then we can add ψ to the end; also we can add χ to the end. *Modus Ponens*: if $(\psi \rightarrow \chi)$ and ψ are both lines in the proof then we can add χ to the proof. See Mates (1972) and Enderton (1972) for presentations of proof rules. There is an algorithm which tells us for any (finite) sequence of formulas whether it is a proof or not.

Theorem 7.12 (Soundness)**.** *For all $S \subseteq SL$, all $\varphi \in SL$, if $S \vdash \varphi$ then $S \models \varphi$.*

Theorem 7.13 (Completeness)**.** *For all $S \subseteq SL$, all $\varphi \in SL$, if $S \models \varphi$ then $S \vdash \varphi$.*

The completeness property of SL tells us that in SL we can syntactically characterize the entailment relation. Whenever $S \models \varphi$ there is a proof from S to φ. Moreover the proof meets the soundness condition that whenever we syntactically derive some ψ from some premises then those premises really entail ψ. Thus in SL we can syntactically characterize the entailment relation.

It is of interest to wonder whether natural languages have the four properties we have adduced in this section. In a later chapter we review these properties for a much richer logic, *First Order Predicate Logic*.

We now take our leave from Sentential Logic and consider the semantic interpretation of a small fragment of English, equipped now conceptually with what we expect a semantic interpretation to be.

7.3 Interpreting a Fragment of English

Interpreting predicates and their arguments is more interesting and more challenging than merely interpreting boolean compounds of Ss. It does however build on the same type of recursive formulations.

Informally first, a model $\mathcal{M}$ consists in part of a set of objects (often called *entities*). This set is noted $E_{\mathcal{M}}$ and called the *domain* or *universe* of the model $\mathcal{M}$. In addition a model must tell us which properties each object has and which relations it bears to the other objects. So a model $\mathcal{M}$ for a language like $\mathcal{L}$(Eng) must tell us what object *Kim* is, what object *Sasha* is, etc. It must tell us which objects are *smiling*, which *crying*, etc. and finally it must tell us which objects are *criticizing* which others, which objects are *praising* which others, etc. Thus,

Definition 7.15. A *model* $\mathcal{M}$ for a language L is a pair $(E_{\mathcal{M}}, [\![\cdot]\!]_{\mathcal{M}})$, where $E_{\mathcal{M}}$ is a non-empty set called the *domain* (or *universe*) of $\mathcal{M}$ and $[\![\cdot]\!]_M$ is a function assigning a denotation to each expression of L.

The function $[\![\cdot]\!]_{\mathcal{M}}$ is defined recursively: denotations are assigned to lexical items with some freedom (as in SL in which the atomic formulas are interpreted freely), and then denotations are assigned compositionally to derived expressions as a function of the denotations assigned to their constituents. Thus a definition of $[\![\cdot]\!]_M$ comes in two parts: (1) specifying its values on the lexical items, and (2) specifying its value recursively on derived expressions (which is what is meant by *compositionality*). We treat these in turn.

7.3.1 Interpreting lexical items

Interpreting NPs and P$_1$s. Given a domain $E_{\mathcal{M}}$, we stipulate that NPs, such as (Kim, NP), (Dana, NP), etc. denote elements of $E_{\mathcal{M}}$. That is, $[\![(s, NP)]\!]_{\mathcal{M}} \in E_{\mathcal{M}}$. And we say that $\textsc{den}_{\mathcal{M}}(\text{NP}) = E_{\mathcal{M}}$, meaning that the set in which expressions of category NP take their denotations in $\mathcal{M}$ is $E_{\mathcal{M}}$. For a language like $\mathcal{L}$(Eng) it is quite arbitrary what element of $E_{\mathcal{M}}$ a particular NP denotes. Different models, even ones with the same domain, can make different choices here.

In general for $\mathcal{M}$ a model and C a category of expression in the language L we are providing models for we write $\textsc{den}_{\mathcal{M}}(C)$ for the set in which expressions of category C are interpreted in $\mathcal{M}$. $\textsc{den}_{\mathcal{M}}(C)$ is called the *denotation set* for C in $\mathcal{M}$. Thus $\textsc{den}_{\mathcal{M}}(\text{NP}) = E_{\mathcal{M}}$ and from what we have said earlier, $\textsc{den}_{\mathcal{M}}(\text{S}) = \{T, F\}$. S is exceptional in that $\textsc{den}_{\mathcal{M}}(S)$ does not vary from model to model, it is always $\{T, F\}$ [2]. However for $\mathcal{M}$ and $\mathcal{N}$ different models $E_{\mathcal{M}}$ and $E_{\mathcal{N}}$ may be different sets. The only general requirement we place on the domain of a model is that it be non-empty (and that requirement is imposed just to streamline certain statements, it could perfectly well be dispensed with). And now, recalling that we write $[A \to B]$ for the set of functions from A into B, we stipulate that:

(21) $\textsc{den}_{\mathcal{M}}(\text{NP}\backslash\text{S})$ is $[\textsc{den}_{\mathcal{M}}(\text{NP}) \to \textsc{den}_{\mathcal{M}}(\text{S})]$, that is,
$[E_{\mathcal{M}} \to \{T, F\}]$.

This is a natural way to represent *properties* of objects in $E_{\mathcal{M}}$. They are functions which look at each object x and say Yes or No, according as x has the property or not. So $[\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{M}} \in [E_{\mathcal{M}} \to \{T, F\}]$. And now we can state the compositional interpretation of a simple S consisting of an NP and a P$_1$. They are combined by the rule of FA to form an S, a truth value denoting expression. The truth value it

[2] Actually it would be more in keeping with the model theoretic approach to simply require (see Chapter 8) that $\textsc{den}_{\mathcal{M}}(\text{S})$ be any two element boolean lattice (they are all isomorphic). The actual objects in the lattice could vary. T would always just be the maximal element of the lattice and F the least element. But we follow tradition here and fix $\textsc{den}_{\mathcal{M}}(\text{S}) = \{T, F\}$, all $\mathcal{M}$.

denotes is the one obtained by applying the P_1 denotation in $\mathcal{M}$ to the NP denotation in $\mathcal{M}$. Thus we require:

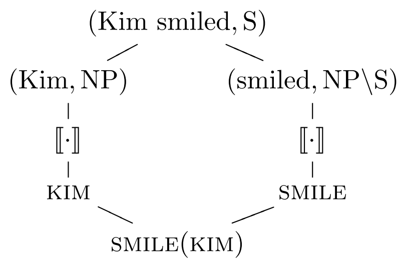
(22) $[\![\text{FA}((\text{Kim}, \text{NP}), (\text{laughed}, \text{NP}\backslash\text{S}))]\!]_{\mathcal{M}}$
$\quad = [\![(\text{laughed}, \text{NP}\backslash\text{S})]\!]_{\mathcal{M}}([\![(\text{Kim}, \text{NP})]\!]_{\mathcal{M}})$.

Here we see explicitly that the interpretation of

$$FA((\text{Kim}, \text{NP}), (\text{smiled}, \text{NP}\backslash\text{S})),$$

which is just (Kim smiled, S), is given in terms of the interpretation of its two immediate constituents, (smiled, NP\S) and (Kim, NP). Noting denotations in upper case for the moment, we can represent the derivation of (Kim smiled, S) by the upper tree below, and its semantic interpretation by the lower tree (with its root at the bottom).

(23)



And the interpretative pattern here is fully general.

(24) In all models $\mathcal{M}$,
$\text{DEN}_{\mathcal{M}}(B\backslash A) = \text{DEN}_{\mathcal{M}}(A/B) = [\text{DEN}_{\mathcal{M}}(B) \to \text{DEN}_{\mathcal{M}}(A)]$.

Thus a slash category expression is interpreted as a function whose domain is the denotation set of the denominator category—the one under the slash—and whose codomain is the denotation set associated with the numerator category. In traditional terms our two semantic primitives are truth and reference. The sets in which expressions denote are either $E_{\mathcal{M}}$, reference, or $\{T, F\}$, truth, or built from them recursively by forming sets of functions.

(25) For all vocabulary strings s, t, all categories A, B and all models $\mathcal{M}$,

a. $[\![\text{FA}((s, B), (t, B\backslash A))]\!]_{\mathcal{M}} = [\![(t, B\backslash A)]\!]_{\mathcal{M}}([\![(s, B)]\!]_{\mathcal{M}})$.

b. $[\![\text{FA}((s, B), (t, A/B))]\!]_{\mathcal{M}} = [\![(t, A/B)]\!]_{\mathcal{M}}([\![(s, B)]\!]_{\mathcal{M}})$.

For example consider two models, $\mathcal{M}$ and $\mathcal{N}$, satisfying the following conditions:

(26) a. $E_{\mathcal{M}} = \{a, b, c\}$ and $E_{\mathcal{N}} = \{b, d, e, f\}$,

b. $[\![(\text{Kim}, \text{NP})]\!]_{\mathcal{M}} = a$, $\quad [\![(\text{Sasha}, \text{NP})]\!]_{\mathcal{M}} = b$,
$[\![(\text{Kim}, \text{NP})]\!]_{\mathcal{N}} = b$, $\quad [\![(\text{Sasha}, \text{NP})]\!]_{\mathcal{N}} = d$.

c.

x	$[\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{M}}(x)$	x	$[\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{N}}(x)$
a	T	b	F
b	F	d	T
c	F	e	F
		f	T

Thus in $\mathcal{M}$, *Kim smiled* is true, but in $\mathcal{N}$ it is false. Formally,

(27) $[\![\text{FA}((\text{Kim}, \text{NP}), (\text{smiled}, \text{NP}\backslash\text{S}))]\!]_{\mathcal{M}}$
$= [\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{M}}([\![(\text{Kim}, \text{NP})]\!]_{\mathcal{M}})$
$= [\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{M}}(a)$
$= T$

(28) $[\![\text{FA}((\text{Kim}, \text{NP}), (\text{smiled}, \text{NP}\backslash\text{S}))]\!]_{\mathcal{N}}$
$= [\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{N}}([\![(\text{Kim}, \text{NP})]\!]_{\mathcal{N}})$
$= [\![(\text{smiled}, \text{NP}\backslash\text{S})]\!]_{\mathcal{N}}(b)$
$= F$

Interpreting $\mathbf{P_2}$s The condition in (24) tells us that the denotation set for P_2 = (NP\S)/NP in a model $\mathcal{M}$ is the set of functions with domain $E_{\mathcal{M}}$ and codomain the denotation set for P_1s, namely $[E_{\mathcal{M}} \to \{T, F\}]$. For example let $\mathcal{M}$ be a model with universe $\{a, b, c\}$. Then $[\![(\text{praise}, P2)]\!]_{\mathcal{M}}$ could be that function PRAISE given by:

(29)

x	PRAISE$(a)(x)$	PRAISE$(b)(x)$	PRAISE$(c)(x)$
a	F	F	F
b	T	T	T
c	F	T	F

In this model b praised everyone, including himself, a didn't praise anyone, and c praised only b but not a or himself.

Exercise 7.11.

a. Exhibit a three element model satisfying:
No one praised himself and no one praised everyone but everyone was praised by someone.

b. Exhibit a model with a three element universe simultaneously satisfying (i) (ii) and (iii):
   i. Someone smiled and everyone who smiled laughed.
   ii. Not everyone laughed.
   iii. Exactly two objects were criticized and they were criticized by different people.

A common alternative notation. Many texts treat P_1 denotations as subsets of the universe E (we omit the subscript $\mathcal{M}$ when not pertinent). So on that approach we say that *John walks* is true in $\mathcal{M}$ iff the object *John* denotes in $\mathcal{M}$ is an element of the set *walks* denotes in $\mathcal{M}$. That approach is equivalent to the one given here. Each subset K of E corresponds to a function CH_K from E into $\{T, F\}$ which maps to T just the elements of K. CH_K is called the *characteristic function* of K. So for anything that can be said about K on the set approach we can formulate a comparable statement about CH_K on the function approach. For example to say that some object $b \in K$ we just say $CH_K(b) = T$. Conversely, the functions g from E into $\{T, F\}$ correspond one to one to the subsets of E. To each such g we associate its *truth set* T_g, namely $\{x \in E \mid g(x) = T\}$. So any statement about g can be translated into a statement about its truth set on the set approach[3].

Similarly the set oriented approach interprets a P_2 as a set of ordered pairs of entities in E. But consider how we presented the function PRAISE in (29). Essentially we interpreted *John praised Mary* by (PRAISE(MARY))(JOHN), in which PRAISE(MARY) is a P_1 function (mapping objects to truth values). The more traditional approach would interpret PRAISE as a set of ordered pairs and *John praised Mary* would be True iff $\langle \text{JOHN}, \text{MARY} \rangle \in$ PRAISE. Obviously enough any statement given in either of these formats can be transformed into one in the other. We prefer our function oriented approach because it provides a denotation for *praised Mary*, which linguists agree is a constituent of *John praised Mary.* The ordered pair approach treats the pair consisting of the subject and the object as denoting a pair of entities, even though they do not form a constituent.

Lexical constraints on interpretations. In natural languages it happens often that lexical items are not interpreted freely in their denotation set: the interpretation of one lexical item may constrain that of another. If one is built from another, e.g. *slowly* from *slow*, we expect by Compositionality that the interpretation of the derived expression is not independent of that of the one from which it is derived. But many morpho-syntactically independent lexical items also exhibit interpretative dependencies. For example:

(30) Antonyms:

a. If Kim is awake then Kim is not asleep.
b. If Kim is male then Kim is not female.
c. If the door is open it is not closed.

Thus acceptable interpretations of lexical items for English cannot freely interpret *awake* and *asleep*, *male* and *female*, etc. Treating them as P_1s for simplicity here we must require of interpretations in a model

[3] In the next chapter we see that $[E \to \{T, F\}]$ and $P(E)$ are boolean lattices which are isomorphic, implying that they make the same sentences true.

that *meaning postulates* like those in (31) hold:

(31) For all $x \in E_{\mathcal{M}}$,

a. if $[\![\text{alive}]\!]_{\mathcal{M}}(x) = T$ then $[\![\text{dead}]\!]_{\mathcal{M}}(x) = F$, and

b. if $[\![\text{male}]\!]_{\mathcal{M}}(x) = T$ then $[\![\text{female}]\!]_{\mathcal{M}}(x) = F$, and

c. if $[\![\text{open}]\!]_{\mathcal{M}}(x) = T$ then $[\![\text{closed}]\!]_{\mathcal{M}}(x) = F$.

The study of these interpretative dependencies is part of *Lexical Semantics*. It covers much more than simple antonyms. Consider for example that *kill* and *dead* are not interpretatively independent: If x *killed* y then y *is dead*. In our formalism we require:

(32) For all models $\mathcal{M}$, if $[\![\text{kill}]\!]_{\mathcal{M}}(y)(x)$ then $[\![\text{dead}]\!]_{\mathcal{M}}(y) = T$.

Finally, while often we have considerable freedom in deciding what element of its denotation set a given lexical item denotes sometimes the denotation is fixed, meaning we have no freedom at all. This is often the case for Determiner denotations. But at the lower level of P_1 and P_2 denotations we have a few candidates. For example we might require that:

(33) For all models $\mathcal{M}$, all $x \in E$, $[\![\text{exist}]\!]_{\mathcal{M}}(x) = T$.

Similarly among the P_2s we find that *is* is not freely interpreted. For example, usually a P_2 does not require that its two NP arguments denote the same entity. While it is quite possible for someone to praise himself, typically an assertion of *John praised Bill* invites the inference that *John* and *Bill* are different people, and in any event they are certainly not required to be the same person. But *is* combines with two NPs to form a sentence and *John is Bill* precisely asserts that *John* and *Bill* are the same individual and doesn't assert anything further. So we might reasonably require of interpretations of English that

(34) For all models $\mathcal{M}$, $[\![\text{is}]\!]_{\mathcal{M}}(y)(x) = T$ iff $x = y$.

This will yield the correct result for *John is Bill*. Once we give Det + N denotations, it also yields without change correct results for *John is a student* and *John is no student* contrary to claims sometimes made in the literature that *is* is ambiguous according as it takes proper nouns like *Bill* or quantified DPs like *a student* as second argument. For most choices of quantified DP however Ss built from *is* are bizarre (though interpretable): *John is every student* implies that there is just one student, John. *John is exactly two students* has to be false, etc.

Lastly here, and curiously, it seems that P_3 present no "logical" members analogous to *exists* among the P_1s or *is* (*to be*) among the P_2s. The presence of an underived verb meaning *give* is quite general (?universal) across languages so the category P_3 is universally available. Yet no language to our knowledge has a P_3 *blik* where *Blik(Dana,Kim,Robin)* would be true iff Dana = Kim = Robin. We return now to models of $\mathcal{L}$(Eng).

Interpreting DPs. We have been treating P_0 as full sentences—they combine with zero NPs to form a sentence. And in general P_{n+1}s combine with one NP to form a P_n. Semantically P_0s denote in $\{T, F\}$ and P_{n+1}s denote functions from E, the domain of the model, into the set of possible P_n denotations. To say this more explicitly let us write $\mathbf{P_n}$ in boldface for the set of P_n denotations in a model with domain E. Formally,

(35) $\mathbf{P_0} = \{T, F\}$ and $\mathbf{P_{n+1}} = [E \to \mathbf{P_n}]$, all $n \geq 0$.

Recall from Chapter 4 that we treated DPs as expressions that combined with P_{n+1}s to form P_ns. So semantically they should denote functions from $n+1$-place predicate denotations to n-place ones. The complete set of such functions (with domain E understood, $|E| > 1$ to avoid degenerate cases) is:

(36) $[\bigcup_{0\leq n} \mathbf{P_{n+1}} \to \bigcup_{0\leq n} \mathbf{P_n}]$.

In fact we don't need all the functions in this set as denotations for DPs in $\mathcal{L}$(Eng). We consider those we do need, starting with proper noun DPs, e.g. (Kim, DP), etc. We first define the notion of an *individual* and then exhibit denotations for proper noun DPs.

Definition 7.16. For all models $\mathcal{M}$, all $b \in E_{\mathcal{M}}$, I_b, the *individual generated by* b, is that function from $\bigcup_{0\leq n} \mathbf{P_{n+1}}$ into $\bigcup_{0\leq n} \mathbf{P_n}$ given by: $I_b(p) = p(b)$.

For example, if *john* (j) and *mary* (m) are elements of $E_{\mathcal{M}}$ then

(37) $I_j(I_m(\textsc{praise})) = I_j(\textsc{praise}(m))$
$= (\textsc{praise}(m))(j)$.

And we must place the following requirement on interpreting functions $[\![\cdot]\!]_{\mathcal{M}}$:

(38) For all models $\mathcal{M}$, if (s, NP) and (s, DP) are both in the Lexicon, then $[\![(s, \mathrm{DP})]\!]_{\mathcal{M}} = I_{[\![(s,\mathrm{NP})]\!]_{\mathcal{M}}}$.

This just guarantees that $[\![(s, \mathrm{DP})]\!]_{\mathcal{M}} = I_b$ iff $[\![(s, \mathrm{NP})]\!]_{\mathcal{M}} = b$ and thus that our two ways of entering proper nouns into the language does not in fact lead to semantic ambiguity.

(39) Let $\mathcal{M}$ be arbitrary with $[\![(\mathrm{smiled}, \mathrm{NP}\backslash\mathrm{S})]\!]_{\mathcal{M}} = \textsc{smile}$ and $[\![(\mathrm{Dana}, \mathrm{NP})]\!]_{\mathcal{M}} = d$. Hence by (38), $[\![(\mathrm{Dana}, \mathrm{DP})]\!]_{\mathcal{M}} = I_d$. Then
a. $[\![FA((\mathrm{Dana}, \mathrm{NP})(\mathrm{smiled}, \mathrm{P_1}))]\!]_{\mathcal{M}}$
$= [\![(\mathrm{smiled}, \mathrm{P_1}))]\!]_{\mathcal{M}}([\![(\mathrm{Dana}, \mathrm{NP})]\!]_M M)$

$= \text{SMILE}(d)$, and

b. $[\![FA((\text{smiled}, \text{P}_1), (\text{Dana}, \text{P}_0/\text{P}_1)]\!]_{\mathcal{M}}$
$= [\![(\text{Dana}, \text{P}_0/\text{P}_1)]\!]_{\mathcal{M}}([\![(\text{smiled}, \text{P}_1)]\!]_{\mathcal{M}})$
$= I_d(\text{SMILE})$
$= \text{SMILE}(d)$

The last step in (b) is from Definition 7.16. Thus when an s of category NP combines with a P_1 we get the same interpretation as when that s of category DP combines with that P_1.

We still must treat the traditionally more difficult case of quantified DPs like *every student*, *most teachers*, etc. but first let us consider the easier case of *modifiers*.

Interpreting modifiers. Eng presents two types of modifiers: manner adverbs such as *joyfully* and *tactfully* of category $\text{P}_1\backslash\text{P}_1$, and adjectives such as *tall* and *clever*, of category N/N.

Interpreting predicate modifiers. By (24) manner adverbs are interpreted by functions from P_1 denotations to P_1 denotations. These functions are chosen from the restricting ones (see Keenan and Faltz (1985)) which guarantees the basic entailment relation illustrated in (4). To define this notion we observe first that the set of possible P_1 denotations in a model $\mathcal{M}$, namely $[E_{\mathcal{M}} \to \{T, F\}]$, possesses a natural partial order, which we note $\leq$ and define by:

Definition 7.17. For all $p, q \in [E_{\mathcal{M}} \to \{T, F\}]$, $p \leq q$ iff for all $b \in E_{\mathcal{M}}$, if $p(b) = T$ then $q(b) = T$.

Theorem 7.14. *$\leq$ as defined above is a reflexive partial order.*

Proof. Clearly $p \leq p$, since if $p(b) = T$ then, trivially, $p(b) = T$. Regarding transitivity, assume that $p \leq q$ and $q \leq r$. We must show that $p \leq r$. For b arbitrary, suppose $p(b) = T$. Then $q(b) = T$ since $p \leq q$, and thus, since $q \leq r$, $r(b) = T$, which is what we desired to show. Regarding antisymmetry suppose $p \leq q$ and $q \leq p$. We must show that $p = q$. We know that p and q are functions with the same domain and codomain, so it suffices to show that they assign each $b \in E_{\mathcal{M}}$ the same truth value. For b arbitrary, suppose first that $p(b) = T$. Then $q(b) = T$ since $p \leq q$. So they have the same value in this case. Suppose now that $p(b) = F$. Then $q(b) = F$, otherwise $q(b) = T$, whence $p(b) = T$, contrary to assumption, since $q \leq p$. This covers all the cases, so p and q are the same function, that is, $p = q$. $\square$

Definition 7.18. Let $(A, \leq)$ be an arbitrary partially ordered set (poset). That is, A is a set and $\leq$ is a partial order on A. Then a function f from A into A is *restricting* iff for all $b \in A$, $f(b) \leq b$.

Exercise 7.12. Let A be a set. Then $(\mathcal{P}(A), \subseteq)$ is a poset.

a. Let B be a subset of A. Define f_B from $\mathcal{P}(A)$ to $\mathcal{P}(A)$ by setting: $f_B(K) = K \cap B$. Prove that f_B is restricting.
b. Let $b \in A$. Define f_b from $\mathcal{P}(A)$ to $\mathcal{P}(A)$ by: $f_b(K) = K - \{b\}$. Prove that f_b is restricting.

Returning now to P_1 modifiers, we require of interpretations $[\![\cdot]\!]_{\mathcal{M}}$ that

(40) For all models $\mathcal{M}$, for all $(s, \mathrm{P}_1\backslash\mathrm{P}_1)$ in $\mathrm{Lex}_{\mathrm{Eng}}$, $[\![(s, \mathrm{P}_1\backslash\mathrm{P}_1)]\!]_{\mathcal{M}}$ is a restricting function (from $[E_{\mathcal{M}} \to \{T, F\}]$ into $[E_{\mathcal{M}} \to \{T, F\}]$).

Imposing condition (40) on interpretations does guarantee the entailment facts in (4):

(41) Suppose that Kim laughed joyfully. We show that it follows that Kim laughed. Let $\mathcal{M}$ be arbitrary, and for simplicity write k for $(\mathrm{Kim}, \mathrm{NP})_{\mathcal{M}}$. Then:

$[\![(\text{Kim laughed joyfully}, \mathrm{S})]\!]_{\mathcal{M}} = T$	Assumption
$[\![(\text{laughed joyfully}, \mathrm{P}_1)]\!]_{\mathcal{M}}([\![(\mathrm{Kim}, \mathrm{NP})]\!]_{\mathcal{M}}) = T$	(25)
$([\![(\text{joyfully}, \mathrm{P}_1\backslash\mathrm{P}_1)]\!]_{\mathcal{M}}([\![(\text{laughed}, \mathrm{P}_1)]\!]_{\mathcal{M}})(k) = T$	(25)
$([\![(\text{laughed}, \mathrm{P}_1)]\!]_{\mathcal{M}}(k) = T$	(40)
$[\![(\text{Kim laughed}, \mathrm{S})]\!]_{\mathcal{M}} = T$	(25)

In a richer fragment of English than that of $\mathcal{L}(\mathrm{Eng})$ we might include P_1 modifiers that are not restricting, though the examples we are aware of all seem to introduce other complications which lie well beyond the scope of this introduction. Still here are a few candidates. Consider *almost* and *nearly*, as they occur in (42).

(42) a. Kim almost failed the exam. b. John nearly fell off his chair.

Clearly these items are not restricting: (42a) does not entail that Kim failed the exam. Indeed it rather suggests that Kim didn't fail. Similarly (42b) does not entail that John fell off his chair. So if we treat almost and nearly as P_1 modifiers they will not denote restricting functions. Syntactically however these expressions differ somewhat from manner adverbs—they naturally occur before the predicate, not after (*??Kim laughed almost*, *??John fell off his chair nearly*) and they seem to assume a much deeper analysis of P_1s than we have offered so far. Namely they introduce a notion of process, whereby an action can be partially but not totally completed. So *Kim almost failed the exam* suggests that Kim took the exam and received a grade that was just good enough to pass.

Another candidate class of non-restricting P_1 modifiers are words like *apparently* and *possibly*, as in (43).

(43) a. Gore apparently won the election.
b. John possibly ran out of bounds.

Apparently (possibly) winning an election does not entail winning it, so these "-ly" adverbs are not restricting. But like *almost* and *nearly* they don't pattern positionally with the manner adverbs.

Interpreting noun modifiers. The category N of nouns is a primitive category, not derived by any of the slash functions, so we stipulate its denotation set:

(44) For all models M, $\text{DEN}_{\mathcal{M}}(\text{N}) = \mathcal{P}(E_{\mathcal{M}})$, the set of subsets of $E_{\mathcal{M}}$.

So a noun such as $(\text{student}, \text{N})$ will be interpreted as a subset of the domain of the model. And we already know that any power set is a poset, partially ordered by the subset relation $\subseteq$. We also know from (24) that adjectives, of category N/N, are functions mapping $\mathcal{P}(E_{\mathcal{M}})$ to $\mathcal{P}(E_{\mathcal{M}})$. Thus it makes sense to ask whether the functions we need to interpret these adjectives are restricting, and they clearly are. A clever student is a student, a female lawyer is a lawyer, etc. So, analogous to (40) we impose the following condition on interpretations in a model:

(45) For all models $\mathcal{M}$, all lexical items $(s, \text{N/N})$, $[\![(s, \text{N/N})]\!]_{\mathcal{M}}$ is a restricting function from $\mathcal{P}(E_{\mathcal{M}})$ to $\mathcal{P}(E_{\mathcal{M}})$.

This condition on interpretations accounts for facts such as those in (46) once expressions of the form Det + N are interpreted:

(46) *Every clever student is a student* is true in all models $\mathcal{M}$; so is *If Kim is a female lawyer then Kim is a lawyer.*

Det + N denotations. DPs such as *every student* and *some lawyer* denote functions from $\bigcup_{0 \le n} \mathbf{P_{n+1}}$ into $\bigcup_{0 \le n} \mathbf{P_n}$. So in particular they map P_1 denotations, $[E \to \{T, F\}]$, into $\{T, F\}$, the set of P_0 denotations. Such functions are called *generalized quantifiers*, GQs. The values that a DP takes at P_n in general is determined by the values it takes at P_1 denotations. So first we concentrate on those, and then show how the extend from P_1 denotations to the full set of P_n denotations. And since Dets like *every*, *some*, etc. combine with Ns like *student* to form such DPs they map each subset of the domain of a model to a generalized quantifier. As we have noted, it is a common property of Dets that they are logical constants—they have a fixed interpretation in each model $\mathcal{M}$. Here is an illustrative example:

(47) For all models $\mathcal{M}$, $[\![(\text{every}, \text{DP/N})]\!]_{\mathcal{M}}$ is that function EVERY which maps each subset A of $E_{\mathcal{M}}$ to that GQ which maps a property p to T iff $A \subseteq \{x \in E \mid p(x) = T\}$.

Writing denotations in upper case and omitting the subscript $\mathcal{M}$, this definition tells us that the interpretation (in M) of (every student laughed, S) is given by:

(48) $(\textsc{every}(\textsc{student}))(\textsc{laugh})$,

and according to (47) this is T iff $\textsc{student} \subseteq \{x \in E \mid \textsc{laugh}(x) = T\}$. Thus *Every student laughed* is true in a model $\mathcal{M}$ iff the set of students in $\mathcal{M}$ is a subset of the set of objects that laughed in $\mathcal{M}$, which is pre-theoretically correct. In this same informal vein we note the denotations of some other Dets invoked in Chapter 4.

(49)
a. $\textsc{some}(A)(p) = T$ iff $A \cap \{x \in E \mid p(x) = T\} \neq \emptyset$
b. $\textsc{no}(A)(p) = T$ iff $A \cap \{x \in E \mid p(x) = T\} = \emptyset$
c. $(\textsc{exactly two})(A)(p) = T$ iff $|A \cap \{x \in E \mid p(x) = T\}| = 2$
d. $(\textsc{the one})(A)(p) = T$ iff $|A| = 1$ and $A \subseteq \{x \in E \mid p(x) = T\}$
e. $\textsc{most}(A)(p) = T$ iff $|A \cap \{x \in E \mid p(x) = T\}| > |A|/2$

We take the indefinite article *a* in *a student* to mean the same as *some*, in *some student*.

These definitions enable us to provide interpretations for subject DPs, ones that combine with a P_1 to form a P_0, as in *Some student laughed loudly*, *Most students read the Times*, etc. Understanding the interpretation of Dets and the subject DPs they build is crucial to understanding their interpretations when they combine with P_2s and P_3s, as well as much in the following chapters, so we advise the reader to work through Exercise 7.13 below immediately in order to build familiarity with these functions.

Exercise 7.13. (Informal models.) Below we simplify notation of some NPs and P_1s by giving them in upper case. E.g. we just say $\textsc{john} = j$, rather than $[\![(\text{john}, \text{NP})]\!]_{\mathcal{M}} = j$, etc.

$\mathcal{M}1$**:** the domain $E_{\mathcal{M}1} = \{a, b, c, d, e\}$

$\textsc{student} = \{a, c, e\}$ $\quad \textsc{adam} = a$

$\textsc{ahtlete} = \{a, b\}$ $\quad \textsc{barry} = b$

x	$\textsc{laugh}(x)$	$\textsc{cry}(x)$	$\textsc{faint}(x)$	$\textsc{smile}(x)$
a	T	F	T	T
b	T	T	F	T
c	F	T	F	T
d	F	T	T	F
e	T	F	F	T

For each sentence below indicate whether it is T or F in this model. If it is F say why. Some of the sentences use constructions we have not yet explicitly given an interpretation for, but which you should be able to figure out.

a. Every athlete laughed.
b. Every athlete cried.
c. Every athlete both laughed and cried.
d. Some athlete both laughed and cried.
e. No athlete cried.
f. Exactly two students laughed.
g. Exactly two students fainted.
h. Exactly one athlete laughed.
i. Most students fainted.
j. Every athlete either laughed or cried.
k. Not every athlete either laughed or cried.
l. No student is an athlete.
m. Barry is Adam.
n. Barry fainted.
o. Some student is an athlete.
p. At least one student both laughed and cried.
q. Not every student smiled.

Object DPs. We consider now how we interpret DPs such as *every teacher* in (50) when they function as objects of P_2s.

(50) Kim praised every teacher.

Every teacher in (50) combines with the P_2 *praised* to form the P_1 *praised every teacher* and so semantically will map the P_2 denotation to a P_1 denotation. In the case at hand we know just what function it is: writing denotations in upper case for simplicity, (EVERY TEACHER) maps PRAISE to that property which holds of an entity b iff "b praised every teacher". That is, iff for every teacher t, $\text{PRAISE}(t)(b) = T$.

And this function is determined by the generalized quantifier that *every teacher* denotes as a subject. Curiously the definition is so simple it is tricky. Let us write, for each $b \in E$, PRAISE_b for that property which maps an entity y to the truth value $\text{PRAISE}(y)(b)$. So PRAISE_b is the property of being praised by b. (For those of you familiar with the lambda notation (Chapter 9), $\text{PRAISE}_b = \lambda x.\text{PRAISE}(x)(b)$). Now we give the value of (EVERY TEACHER) at PRAISE:

(51) $(\text{EVERY TEACHER})(\text{PRAISE})(b) = (\text{EVERY TEACHER})(\text{PRAISE}_b)$.

The right hand side of the = sign in (51) uses (EVERY TEACHER) as a function from properties to truth values. Given that function its value at a P_2 denotation like PRAISE is uniquely determined, as in (51). To see that this definition is the right one, consider the truth conditions of the right hand side of (51).

(52) $(\text{EVERY TEACHER})(\text{PRAISE}_b) = T$

iff $\text{TEACHER} \subseteq \{x \mid \text{PRAISE}_b(x) = T\}$	Def EVERY
iff for each $t \in \text{TEACHER}$, $t \in \{x \mid \text{PRAISE}_b(x) = T\}$	Def $\subseteq$
iff for each $t \in \text{TEACHER}, \text{PRAISE}_b(t) = T$	Set Notation
iff for each $t \in \text{TEACHER}, \text{PRAISE}(t)(b) = T$	Def PRAISE_b

And this last line says just what we want it to say: *praised every teacher* denotes that property which is true of an entity b just in case for every teacher t, b praised t.

The important point about the definitions above is that a map from P_1 denotations to P_0 denotations uniquely determines a (narrow scope) map from P_2 denotations to P_1 denotations (and more generally from P_{n+1} denotations to Pn denotations).

Definition 7.19.

a. For P a possible P_2 denotation (a map from entities to properties) and $b \in E$, write P_b for that property defined by: $P_b(y) = P(y)(b)$.
b. For F a generalized quantifier (a map from properties to truth values) we extend F to a function from P_2 denotations to properties as follows: $F(P)(b) = F(P_b)$.

Here P is possible P_2 denotation, so $F(P)$ is a property, the one whose value at an arbitrary object b is whatever truth value F assigns to the property P_b. In this way we see that the value of a DP denotation at P_2 denotations is uniquely determined once we have given its values at the possible P_1 denotations. No additional interpretative apparatus such as Geach Division is needed.

This question has provoked much discussion in the semantics literature (van Benthem, 1986, Ch. 7), Heim and Kratzer (1998), Keenan (1989), Montague (1970) so let us show that the form in Definition 7.19 above applies to P_{n+1}s in general (though we do not use the added generality here). The denotation P of a P_{n+1} maps n entities $b_1, \ldots, b_n$ in succession to a property. And analogous to Definition 7.19a, for $b = (b_1, \ldots, b_n)$ an n-tuple of entities in E^n let us write P_b for that property which maps an object y to the truth value $P(y)(b_1), \cdots, (bn)$.

Then DPs have just one denotation—one whose domain is the set of $n+1$-ary predicate denotations and whose values are n-ary predicate denotations, as in Keenan (1992) and Keenan and Westerståhl (1996). For the formal record, given E, we define:

Definition 7.20.
$\text{Den}_E(\text{DP}) = \{F \in [\bigcup_n \mathbf{P_{n+1}} \to \bigcup_n \mathbf{P_n}]$
$\mid \text{for all } n, \text{all } P \in \mathbf{P_{n+1}}, \text{all } b \in E^n, F(P)(b) = F(P_b)\}$
Where for $b \in E^n$, and $P \in \mathrm{P}_{n+1}$, P_b is that element of $[E \to \{T, F\}]$ given by: $P_b(a) = P(a)(b_1)\cdots(b_n)$.

In this way then a DP such as *every student* does not have many denotations, it just has one, a function with a large domain. Similarly we can now understand DP as a single category, the category of expression which combines with P_{n+1}s to form Pns, all n.

We note though that the interpretation we obtain for object DPs is the object narrow scope one. To get the object wide scope reading it will be helpful to use lambda abstraction (Chapter 9), but it is worth noting that object wide scope readings are often, in practice, not available (see Beghelli et al. (1997)). Consider (53).

(53) a. Each student answered no question correctly on the exam.
b. Fewer than five students answered every question correctly.

The object wide scope (OWS) reading of (53a) says that no question has the property that each student answered it correctly. But in fact speakers do not use (53a) with that meaning. It only has the stronger if less probable reading that each student missed every question. Similarly in (53b) the OWS reading says that every question has the property that fewer than five students answered it correctly. But in fact (53b) just means that the number of students who got a perfect score was less than five. The reading of such Ss that our analysis to date does capture is by far the most natural one. The less natural OWS reading requires a richer interpretative apparatus (Chapter 9).

Let us work though one example to see that we do represent the object narrow scope reading (and also just to see how the mechanism of interpretation works with multiply-quantified Ss). Consider (54) from $\mathcal{L}(\text{Eng})$.

(54) Some student praised every teacher.

This is derived by FA applied to $(\text{praised every teacher}, \mathrm{P}_1)$ and $(\text{some student}, \mathrm{P}_0/\mathrm{P}_1)$. By our semantics for some (49a) this is interpreted as T iff

(55) $\textsc{student} \cap \{x \mid [\![\text{praised every teacher}]\!]_{\mathcal{M}}(x) = T\} \neq \emptyset$
iff $\textsc{student} \cap \{x \mid (\textsc{every teacher})(\textsc{praise})(x) = T\} \neq \emptyset$
iff $\textsc{student} \cap \{x \mid (\textsc{every teacher})(\textsc{praise}_x) = T\} \neq \emptyset$

iff $\text{STUDENT} \cap \{x \mid \text{TEACHER} \subseteq \{y \mid (\text{PRAISE}_x)(y) = T\}\} \neq \emptyset$
iff $\text{STUDENT} \cap \{x \mid \text{TEACHER} \subseteq \{y \mid \text{PRAISE}(y)(x) = T\}\} \neq \emptyset$
iff there is a $b \in \text{STUDENT}$ such that
$\text{TEACHER} \subseteq \{y \mid \text{PRAISE}(y)(b) = T\}$.

This last line just says that there is a student who is such that the set of things he praised includes all the teachers. That is, every teacher has narrow scope in (54).

A concluding speculation

Treating the sequence of n-place predicates as a single object (the argument of DPs) is somewhat novel. It is natural to wonder whether there are other structure building operations which avail themselves of this generalization. Here are a few prima facie plausible suggestions.

Locative PPs as $\mathbf{P}_n$ modifiers (Keenan (1981), Keenan and Faltz (1985)). Consider the entailment paradigms in (56)–(59) below. In all cases the source locative *from the attic* predicates a location of one or another argument, the subject in (56a), whence the entailment of (56b). But in (57) it is the object argument and not the subject which the location is predicated of.

(56) a. John sang / shouted / fell from the attic.
b. $\models$ John was in the attic.

(57) a. John took /removed / withdrew the trunk from the attic.
b. $\models$ The trunk was in the attic.
c. $\not\models$ John was in the attic.

These data can be accounted for if we treat *from the attic* as combining with either a P_1 or a P_2 and predicating location of its argument. In (58) we just treat *from the attic* as combining with the complex P_1 *watched Bill* (*attacked Bill*, etc.). So the PP is sensitive to the type of P_2; it can tell the difference between *take*, *remove*, etc. on the one hand and *watch*, *attack*, etc. on the other.

(58) a. John watched / attacked / studied Bill from the attic.
b. $\models$ John was in the attic.
c. $\not\models$ Bill was in the attic.

And finally, in (59) we can represent the ambiguity according as *from the attic* combines with the P_2 *shoot*, *grab*, etc. thus predicating of Bill, or it combines with the complex P_1 *shot Bill*, *grabbed Bill*, etc. predicating of its subject John.

(59) John shot / grabbed / called Bill from the attic.
(Ambiguous according as John or Bill was in the attic. If it was Bill then he is understood to have moved or be intended to

move from the attic).

Passives. A variety of languages present a strict morphological passive formed by affixing the active form of the verb (see Keenan and Dryer (2007)). The derived, passive, predicate has one fewer arguments than active one it is built from. In the most widely cited cases the active verb is a P_2 and the passive one a P_1. However languages which have such passives—Latin, Turkish, Kinyarwanda (and Eastern Bantu quite generally) also always allow passive morphology on P_3s forming P_2s. So if a language can say *The door was opened* it can also say either *Sue was given the key* or *The key was given to Sue* (and perhaps both). Less well known is that in some of these languages passive morphology also applies to P_1s creating P_0s. Turkish (Perlmutter and Postal, 1983), Özkaragöz (1986) and Latin are examples. (In Turkish the passive suffix is *-ın* following laterals, *-n* after vowel-final stems, and *-ıl* elsewhere. Vowels exhibit front-back harmony surfacing as *-ül* or *-ün*.)

(60) Turkish:

a. Active P_2:

Hasan	*bavul+u*	*açtı*
Hasan	suitcase+acc	open+past

'Hasan opened the suitcase.'

b. Passive $P_2 \Rightarrow P_1$:

Bavul	*(Hasan*	*tarafindan)*	*aç-ıl-tı*
suitcase	(Hasan	by)	open+pass+past

'The suitcase was opened (by Hasan).'

c. Passive $P_1 \Rightarrow P_0$:

Burada	*düş-ü l-ür*
here	fall-pass-aorist

'Here one falls.'

d. Passive twice: $P_2 \Rightarrow P_0$:

Bu	*oda-da*	*döv-ül-ün-ü r*
this	room-loc	hit-pass-pass-aorist

'One is beaten (by one) in this room.'

Causatives It is generally recognized Comrie (1985) that there are languages with a causative affix that derives P_2s from P_1s, and often also applies to P_2s to derive P_3s. (61) and (62) illustrate these ordinary cases from Malagasy (Austronesian). It is less common but still attested that a given causative affix may iterate at least once, deriving a P_3 from a P_1. (63) from Tsez (Daghestanian; Comrie (2000)) is illustrative.

(61) a. *Mihomehy izy ireo*
laugh 3nom dem+pl
'They are laughing.'

b. *Mampihomehy azy ireo aho*
make+laugh 3acc dem+pl 1s.nom
'I am making them laugh.'

(62) a. *manenjika ny ankizy Rabe*
chase the children Rabe
'Rabe is chasing the children.'

b. *mampanenjika an-dRabe ny ankizy aho*
make+chase acc-Rabe the children 1s.nom
'I am making Rabe chase the children.'

(63) *učitel-ā uži-q kidb-eq kec'*
teacher-erg boy-poss girl-poss song-abs
q'aλi-r-er-si
sing-caus-caus-past+witnessed
'The teacher made the boy make the girl sing a song.'

We turn now to a generalized treatment of the boolean connectives and the surprisingly extensive role played by boolean lattices in natural language semantics.

8

Semantics II: Coordination, Negation and Lattices

This chapter provides a unified interpretation for expressions built from the boolean connectives *(both) and*, *(either) or*, *neither nor*. This task is challenging since these connectives are properly *polymorphic*: they combine with expressions in almost any (content) category to form further expressions in that category. We provide a semantic basis for polymorphism and speculate on a deeper explanation for it. We first review (at the risk of repetition) the extensive variety of categories which host coordination:

(1) P_0 (Sentence) Coordination:
   a. Either John came early or Mary stayed late.
   b. Neither did John come early nor did Mary stay late.
   c. Kim insulted Dana and Dana insulted Kim.

(2) P_1 Coordination:
   a. Kim bought a puppy and either laughed or cried.
   b. He neither laughed nor cried.

(3) P_2 Coordination:
   a. Kim either praised or criticized each student.
   b. Kim neither praised nor criticized each student.

(4) P_3 Coordination:
   a. Jim either gave or sold Mary his watch.
   b. She neither showed nor handed me the jewels.

(5) CP (Complementizer Phrase) Coordination:
   a. Kim believes either that there is life on Mars or that there isn't (life on Mars).
   b. Ted thinks (both) that the election was rigged and that the state is corrupt.

(6) DP Coordination:

a. Most poets and all musicians daydream often.
b. Either Kim or some student insulted every teacher.
c. Kim interviewed half the male candidates and all the female ones.
d. Neither every student nor every teacher attended the meeting.

(7) Det Coordination:

a. Most but not all students read the Times.
b. John interviewed either exactly two or exactly three candidates.

(8) AdvP (Adverb Phrase) Coordination:

a. John drives rapidly and recklessly.
b. He drives neither rapidly nor recklessly.
c. He works slowly but not carefully.

(9) Preposition Coordination:

a. She lives neither in nor near New York City.
b. The water was flowing over, under and around the car.

(10) PP (Prepositional Phrase) Coordination:

a. He was either at the office or on the train when the accident occurred.
b. He works with Martha and with Bill but not with Ann.

(11) AP (Adjective Phrase) Coordination:

a. No intelligent or industrious student.
b. An attractive but not very well built house.

Question Do these diverse uses of *and* (*or*, *neither...nor...*) have anything semantic in common? Is there any reason to expect boolean connectives to be polymorphic?

For example, surely the meaning of *or* when it combines P_2s is not completely different from its meaning when it combines DPs, etc. Our task in this chapter is to show just what the different usages have in common, answering both parts of the query and exhibiting a non-obvious, possibly deep, generalization about natural language.

8.1 Coordination: Syntax

We assume here essentially the Coordination rule of the previous chapter.

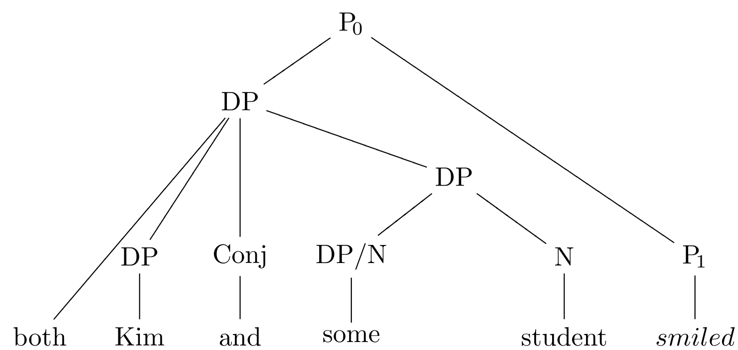
(12) Coord :

$$(s, \mathrm{Conj})(t, C)(u, C) \longrightarrow \begin{cases} (\mathrm{both} \frown t \frown s \frown u, C) & \text{if } s = \text{and} \\ (\mathrm{either} \frown t \frown s \frown u, C) & \text{if } s = \text{or} \\ (\mathrm{neither} \frown t \frown s \frown u, C) & \text{if } s = \text{nor}, \end{cases}$$

where C is any of the categories in (1–11), henceforth called *coordinable* categories.

Here is a sample derivation of an expression.

(13)



In general we do not need to mark which rules applied as that information is recoverable: the rules (structure building functions) of Eng have the property that if a rule F maps a tuple of expressions to an expression u then F is the only rule that maps that tuple to u.

Remarks on syntax.

1. *both*, *either*, and *neither* are not assigned categories on this syntax; they are introduced *syncategorematically* by the rules.
2. We add $(\mathrm{himself}, \mathrm{P}_2\backslash\mathrm{P}_1)$ and $(\mathrm{herself}, \mathrm{P}_2\backslash\mathrm{P}_1)$ to $\mathrm{Lex}_{\mathrm{Eng}}$.

Exercise 8.1. Provide syntactic analysis trees for each of the following.

a. Either Kim or Sasha laughed.
b. Dana criticized both herself and every teacher.

A typological regularity. In examples we often omit *both* and *either* for simplicity. But a two part expression of coordination is not uncommon; often we just repeat the conjunction, as in Russian and French *et Jean et Marie* “and John and Mary”, *ou Jean ou Marie* “or John or Mary”, and *ni Jean ni Marie* “neither John nor Mary”. This is the normal order in V-initial and SVO languages. In V-final languages the order is postpositional: *John-and Mary-and*, as in (14) from Tamil (Corbett (1991) pg. 269).

(14) *raaman-um* *murukan-um* *va-nt-aaka*
Raman-and Murugan-and come+past+3.pl.rational
'Raman and Murugan came.'

Exercise 8.2. The subject DP in *Mary and Sue or Martha can read Greek* is logically ambiguous. Using *both* and *either*, exhibit two DPs each of which unambiguously represents a different interpretation of the subject DP. Describe in words a situation in which one of the Ss they build is true and the other false.

8.2 Coordination: Semantics

Here we provide an answer our initial Question by showing that the sets in which expressions in coordinable categories denote are ones with a particular kind of partial order, a (*Boolean*) *lattice* order. And the core generalization we seek is that no matter what the category of expression coordinated, a conjunction of expressions always denotes the *greatest lower bound* of the denotations of its conjuncts, and a disjunction of expressions denotes the *least upper bound* of its disjuncts. So *and* is a greatest lower bound operator, *or* a least upper bound operator. We now define these notions.

Definition 8.1. For $(L, \leq)$ a poset (L always assumed non-empty), then for $x \in L$ and $K \subseteq L$,

a. i. x is a *lower bound* (lb) for K iff for all $y \in K$, $x \leq y$.
ii. x is a *greatest lower bound* (glb) for K iff x is a lb for K and for all lb's y for K, $y \leq x$.

b. i. x is an *upper bound* (ub) for K iff for all $y \in K$, $y \leq x$.
ii. x is a *least upper bound* (lub) for K iff x is an ub for K and for all ub's y for K, $x \leq y$.

Proposition 8.1. *If a subset K of a poset has glb it has just one.*

Proof. Let x and x' be glb's for K. Then they are both lb's for K and $x \leq x'$ since x' is a greatest lower bound. Similarly since x is a glb we have that $x' \leq x$. So by antisymmetry, $x = x'$. □

Notation If K has a glb it is noted $\bigwedge K$, read as "meet K" or the *infimum* (inf) of K. $\bigwedge\{x, y\}$ is usually written $(x \wedge y)$, read as "x meet y". Note: $(x \wedge y)$ is an element of the poset; trivially $(x \wedge y) \leq x$ since $(x \wedge y)$ is a lb for $\{x, y\}$.

Exercise 8.3. Prove that if a subset K of a poset has a lub it is unique (i.e. it has just one).

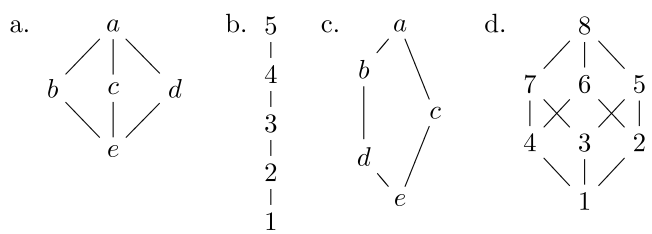
Notation If K has a lub it is noted $\bigvee K$ and read "join K" or the *supremum* (sup) of K. $\bigvee\{x, y\}$ is usually written $(x \vee y)$, read as "x join y".

Two important logical abbreviations. Expressions such as "For all $x \in A, \varphi$", as in the right hand side of Definition 8.1a-i and b-i, are short for "For all x, if $x \in A$ then φ". Thus every element x of the poset L above is a lower bound for $\emptyset$ since this just requires that for all $y \in \emptyset$, $x \leq y$. That is, "for all y, if $y \in \emptyset$ then $x \leq y$" is true, since for each y the antecedent $y \in \emptyset$ is false, so the if-then claim is (vacuously) true. In contrast, when we say "for some $x \in A, \varphi$" we are stating a conjunction not a conditional claim. So a claim of the form "For some $x \in \emptyset, \varphi$" means "There is an $x \in \emptyset$ and φ" and this claim is false, no matter what sentence φ is, since the first conjunct, $x \in \emptyset$, is false.

Definition 8.2. A *lattice* is a partially ordered set $(L, \leq)$ which satisfies for all $x, y \in L$, $\{x, y\}$ has a greatest lower bound and $\{x, y\}$ has a least upper bound.

These conditions may be given by saying "for all $x, y \in L$, $x \wedge y$ and $x \vee y$ exist".

Small lattices are often represented by their Hasse (pronounced: "Hassuh") diagrams, as (15a–d) below. In such diagrams a point (node, vertex) x is understood to bear the lattice order $\leq$ to a point y iff either x is y or you can move up from x along edges and get to y. (15a) is called the *diamond* lattice, (15b) a *chain* lattice, (15c) the *pentagon* lattice, and (15d) is, up to isomorphism, a power set lattice.

(15)


Exercise 8.4. Compute the meets and joins for the lattices as indicated

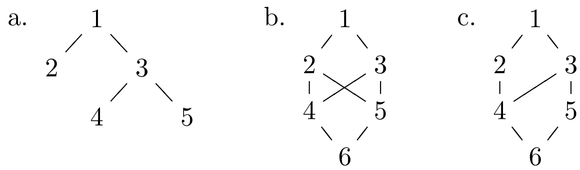
(15a) i. $(b \wedge d)$ ii. $(a \wedge c)$ iii. $b \wedge (c \wedge d)$ iv. $\bigvee\{e, b, c, d\}$

(15b) i. $(4 \wedge 4)$ ii. $\bigwedge\{3\}$ iii. $2 \vee \bigwedge\{5, 3, 4\}$ iv. $((5 \vee 4) \vee 3) \vee 4$

(15c) i. $b \wedge (d \vee c)$ ii. $(e \vee c) \wedge d$ iii.$(d \vee c \vee b) \wedge d$ iv. $\bigvee \emptyset$

(15d) i. $8 \wedge 6 \wedge 3$ ii. $(7 \wedge (3 \vee 2)) \vee 6$ iii. $(7 \wedge 3) \vee (7 \wedge 1)$ iv. $\bigwedge \emptyset$
v. $\bigwedge\{7, \bigvee\{7, (7 \wedge 2)\}\}$ vi. $\bigvee\{1, 2, 3, 4, 5, 6, 7, 8\}$
vii. $\bigwedge\{1, 2, 3, 4, 5, 6, 7, 8\}$

Exercise 8.5. For each Hasse diagram below say whether it is a lattice; if not, give a reason.



Exercise 8.6. Is $(\mathbb{N}, \leq)$ a lattice (where $\leq$ is the ordinary $\leq$ in arithmetic)? If so, what is $m \wedge n$ and $m \vee n$, any $m, n \in \mathbb{N}$?

8.2.1 Some important examples of lattices.

Proposition 8.2. *For A any set, $(\mathcal{P}(A), \subseteq)$ is a lattice.*

Proof. Clearly for all $X \in \mathcal{P}(A)$, $X \subseteq X$, so $\subseteq$ is reflexive. And for $X, Y, Z \in \mathcal{P}(A)$, if $X \subseteq Y$ and $Y \subseteq Z$ then $X \subseteq Z$, so $\subseteq$ is transitive. Finally for $X, Y \in \mathcal{P}(A)$, if $X \subseteq Y$ and $Y \subseteq X$ then $X = Y$ since neither has a member the other doesn't. So $(\mathcal{P}(A), \subseteq)$ is a poset. Now, let $X, Y \in \mathcal{P}(A)$, we must show (a) and (b):

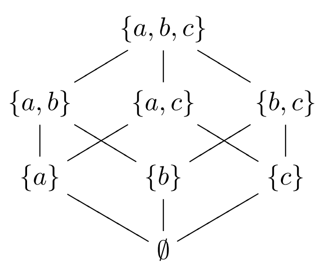
a. $\{X, Y\}$ has a glb in $\mathcal{P}(A)$. But since $X, Y \in \mathcal{P}(A)$, $X \cap Y \in \mathcal{P}(A)$. Clearly $X \cap Y \subseteq X$ and $X \cap Y \subseteq Y$, so $X \cap Y$ is a lb for $\{X, Y\}$. Now let Z be a lower bound for $\{X, Y\}$. Then $Z \subseteq X$ and $Z \subseteq Y$, so any element $z \in Z$ is an element of X and an element of Y and thus an element of $X \cap Y$. Since z was arbitrary, $Z \subseteq X \cap Y$, which is to say that $X \cap Y$ is the greatest lower bound of $\{X, Y\}$, as was to be shown.

b. $X \cup Y$ is the least upper bound of $\{X, Y\}$. □

Exercise 8.7. Prove (b) above.

Thus our familiar power sets are posets with additional structure, a lattice structure. Here for example is the Hasse diagram of $\mathcal{P}(\{a, b, c\})$:

(16)



An important abstraction step. Earlier we defined intersection and union standardly in terms of set membership. We have now characterized those notions in a purely order theoretic way: $X \cap Y = X \wedge Y$, the greatest lower bound of $\{X, Y\}$ in $\mathcal{P}(A)$, and $X \cup Y = X \vee Y$, the least upper bound of $\{X, Y\}$. The whiff of generalization is in the air.

Proposition 8.3. *$(\{T, F\}, \leq)$ is a lattice, defined by the Hasse diagram below:*

$$\begin{array}{c} T \\ | \\ F \end{array}$$

So we understand here that $T \leq T$, $F \leq F$ and $F \leq T$. But $T \not\leq F$. This two element lattice is often represented as $\{0, 1\}$, using 0 for F and 1 for T. It is called the lattice 2.

We verified earlier that $\leq$ is a reflexive partial order, the implication order. But not only is $(\{T, F\}, \leq)$ a poset, it is a lattice. For all $x, y \in \{T, F\}$ the greatest lower bound of $\{x, y\}$ is $x \wedge y$ and its least upper bound is $x \vee y$.

(17)

x	y	$x \wedge y$	$x \vee y$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Reading the second line of (17) let us verify that F is the glb of $\{T, F\}$. Clearly F is a lower bound, since for all $x \in \{T, F\}$, $F \leq x$. Now suppose that some $z \in \{T, F\}$ is a lb for $\{T, F\}$. Then in particular $z \leq F$, showing that F is greatest of the lower bounds for $\{T, F\}$.

Useful remark. For $x, y \in \{T, F\}$, to show that $x \leq y$ it suffices to show that if $x = T$ then $y = T$, since if $x = F$ then $x \leq y$ no matter what y is. So there is only one case to consider. Similarly to show that a statement of the form "if P then Q" is true it suffices to consider the case where P is True. Then we must show that Q is true. If P is False then the statement "if P then Q" is vacuously true. "If P then Q" is false only when P is True and Q False. The conditional statement "if P then Q" is often symbolized $(P \rightarrow Q)$.

A fundamental abstraction. The glb column in (17) gives the standard truth table for conjunction! A sentence of the form (P and Q) is interpreted as True iff both P and Q are. In all other cases (P and Q) is interpreted as False. Similarly the lub table gives the truth table for disjunction: A disjunction (P or Q) is interpreted as True iff either P is or Q is (possibly both). Thus, for P and Q sentences and $\mathcal{M}$ a model,

(18) a. $[\![(P \text{ and } Q)]\!]_{\mathcal{M}} = [\![P]\!]_{\mathcal{M}} \wedge [\![Q]\!]_{\mathcal{M}}$
b. $[\![(P \text{ or } Q)]\!]_{\mathcal{M}} = [\![P]\!]_{\mathcal{M}} \vee [\![Q]\!]_{\mathcal{M}}$

So (18) says that a conjunction of Ss always denotes the glb of the denotations of its conjuncts; a disjunction of Ss denotes their lub. And this is the property of *and* and *or* that generalizes to other categories and answers our initial Question:

(19) For P and Q expressions of any coordinable category C, the equations in (18) hold.

We must of course state what lattices are the denotation sets for the other categories C. Almost all the other cases are covered by:

Proposition 8.4. *If $(A, \leq_A)$ is a lattice and B a set, $([B \to A], \leq)$ is a lattice, with $\leq$ defined by: $f \leq g$ iff for all $b \in B, f(b) \leq_A g(b)$. Such lattices are said to be defined* pointwise.

Proof. Let us see that $([B \to A], \leq)$ is, in fact, a lattice. First we must show that $\leq$ as defined is a partial order. Clearly for all $f \in [B \to A]$, $f \leq f$ since for all $b \in B, f(b) \leq_A f(b)$ because $\leq_A$ is reflexive. In a similar way the transitivity and antisymmetry of $\leq$ is inherited from $(A, \leq_A)$. To show the existence of meets let us define for all functions $f, g \in [B \to A]$ a function $h_{f,g}$ from B into A by setting:

$$h_{f,g}(b) = f(b) \wedge g(b).$$

We claim that $h_{f,g}$ is the glb of $\{f, g\}$. Clearly $h_{f,g} \leq f$ since for all b, $h_{f,g}(b) = f(b) \wedge g(b) \leq_A f(b)$. Similarly $h_{f,g} \leq g$, so $h_{f,g}$ is a lower bound for $\{f, g\}$. To see that it is greatest of the lower bounds, let k be a lb for $\{f, g\}$. So $k \leq f$ and $k \leq g$, so for all b, $k(b) \leq_A f(b)$ and $k(b) \leq_A g(b)$, whence $k(b)$ is a lower bound for $\{f(b), g(b)\}$, so $k(b) \leq_A f(b) \wedge g(b) = h_{f,g}(b)$. So $k \leq h_{f,g}$, whence $h_{f,g}$ is greatest of the lower bounds for $\{f, g\}$. So $h_{f,g} = f \wedge g$ as was to be shown. □

Notational simplification. In noting pointwise lattices $([B \to A], \leq)$ we can usually omit the subscript on $\leq_A$ since in writing $x \leq y$ we know that if $x, y \in A$ then $x \leq_A y$ is intended.

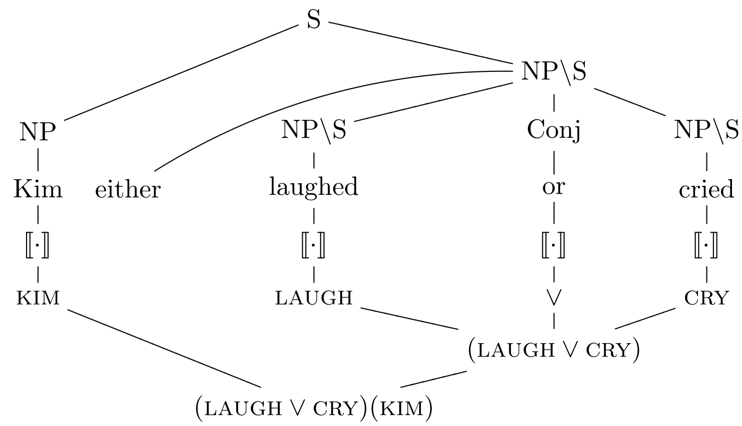
Exercise 8.8.

a. Prove that $\leq$ in $([B \to A], \leq)$ is transitive.
b. Prove that $\leq$ in $([B \to A], \leq)$ is antisymmetric.
c. Prove that for all $f, g \in [B \to A]$, $\{f, g\}$ has a least upper bound.

8.2.2 The use of pointwise lattices in semantics.

Given a model $\mathcal{M}$ with domain E (we omit the subscript on E), the denotation set for expressions of category NP\S = P_1 was given as $[E \to \{T, F\}]$. But this set is the domain of a pointwise lattice, since $\{T, F\}$ is a lattice. And conjunctions of P_1s are interpreted by (19) as the glb of the conjuncts, and disjunctions as the lubs of their disjuncts.

(20)



$= \textsc{laugh}(\textsc{kim}) \vee \textsc{cry}(\textsc{kim})$

Thus we have shown that in each model $\mathcal{M}$ *Kim laughed or cried* and *Kim laughed or Kim cried* are interpreted as the same truth value.

Note that the semantic computation in (20) only works because *Kim* denotes an element of E. Had we used for example *every student* instead of *Kim* then the next to the last line would be ($\textsc{every student}$)($\textsc{laugh} \vee \textsc{cry}$) since P_1 denotations lie in the domain of *every student*, of category P_0/P_1. *Every student* does not denote an element of E and thus does not lie in the domain of ($\textsc{laugh} \vee \textsc{cry}$) so replacing KIM by EVERY STUDENT in the last two lines of (20) is nonsense.

Moreover our semantics from Chapter 7 supports that in general (EVERY A) does not map $(P \vee Q)$ to (EVERY A)$(P) \vee$ (EVERY A)(Q). Observe first that this claim accords with our semantic intuitions of entailment based on ordinary English. Compare:

(21) a. Every student either laughed or cried.
b. Either every student laughed or every student cried.

Imagine a model with 5 students, three laughed but didn't cry and the other two cried but didn't laugh. In such a case (21a) is true: no matter what student you pick, that student either laughed or cried. But (21b) is false; it is not true that every student laughed, and it is not true that every student cried. And to see that our semantics guarantees this result consider that (21a) is interpreted as in (22) below:

(22) (EVERY(STUDENT))(LAUGH OR CRY)
$= T$ iff $\textsc{student} \subseteq \{x \in E \mid (\textsc{laugh or cry})(x) = T\}$
$= T$ iff $\textsc{student} \subseteq \{x \in E \mid \textsc{laugh}(x) \vee \textsc{cry}(x) = T\}$
$= T$ iff for each $x \in \textsc{student}$, x laughed or x cried.

And this last statement can be true in a situation in which just some of the students laughed and the others cried. But in contrast (21b) is a disjunction of Ss and is true if and only if one of the disjuncts is true. The first disjunct says that all the students laughed, the second that they all cried. As both conditions fail in the scenario given above it is false in some models in which (21a) is true, hence (21a) does not entail (21b).

Exercise 8.9. Exhibit an informal model in which sentence (a) below is true and sentence (b) is false. Say why it is false and conclude that (a) does not entail (b). Again this follows on our semantics for SOME given earlier plus that of conjunctions of P_1s given here.

a. Some student laughed and some student cried.
b. Some student both laughed and cried.

The bottommost line in (20) represents directly the denotation of *Either Kim laughed or Kim cried*. Thus (23a,b) below are logically equivalent, where we define:

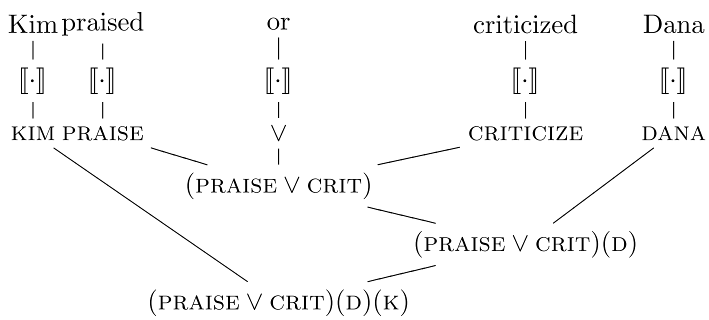
Definition 8.3. Expressions s and t are *logically equivalent* iff for each model $\mathcal{M}$ they have the same denotation in $\mathcal{M}$ (that is, $[\![s]\!]_{\mathcal{M}} = [\![t]\!]_{\mathcal{M}}$).

(23) a. Kim either laughed or cried.
b. Either Kim laughed or Kim cried.

Exercise 8.10. Analogous to (20) exhibit the semantic interpretation trees for (a) and (b) below, concluding that they too are logically equivalent.

a. Dana both laughed and cried.
b. Dana laughed and Dana cried.

Now, once we have seen that $\text{DEN}_{\mathcal{M}}(\mathrm{NP}\backslash\mathrm{S})$ is a (pointwise) lattice we can infer that $\text{DEN}_{\mathcal{M}}(\mathrm{P}_2) = \text{DEN}_{\mathcal{M}}((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP})$ is as well, as it is given as the set $[E \to [E \to \{T, F\}]]$, the set of functions from E into a lattice. (And in general $\text{DEN}_{\mathcal{M}}(\mathrm{P}_{n+1}) = [E \to \text{DEN}_{\mathcal{M}}(\mathrm{P}_n)]$ is a lattice pointwise). In the case of P_2s observe the semantic computation in (24), again crucially using proper NPs not arbitrary DPs.

(24) 

$$= (\text{PRAISE}(\text{DANA}) \vee \text{CRITICIZE}(\text{DANA}))(\text{KIM})$$
$$= \text{PRAISE}(\text{DANA})(\text{KIM}) \vee \text{CRITICIZE}(\text{DANA})(\text{KIM})$$

The last two lines just multiply out the pointwise definition of join at the P_2 and P_1 levels. And as the last line is the interpretation of (25b) below we see that our semantics shows that (25a,b) are logically equivalent.

(25) a. Kim either praised or criticized Dana.
b. Either Kim praised Dana or Kim criticized Dana.

We see then that the denotation set for a slash category A/B or $B\backslash A$ assumes pointwise the lattice structure of $\text{DEN}_{\mathcal{M}}A$, the set in which the functions take their values.

(26) **Functional Projection (FP)**: For all categories A, B, $\text{DEN}_{\mathcal{M}}(A/B) = \text{DEN}_{\mathcal{M}}(B\backslash A) = [\text{DEN}_{\mathcal{M}}(B) \rightarrow \text{DEN}_{\mathcal{M}}(A)]$, assumes the structure of $\text{DEN}_{\mathcal{M}}(A)$ pointwise.

Functional Projection covers many more cases than just the n-place predicates. A basic case is P_0/P_1, that is, S/(NP\S) in unabbreviated form. Now $\text{DEN}_{\mathcal{M}}(P_0)$ is a lattice so we take $\text{DEN}_{\mathcal{M}}(P_0/P_1)$ to be the pointwise lattice built from it. This guarantees logical equivalences like the (a,b) pairs below (which shows that Boolean compounds of DPs "distribute" over P_ns):

(27) a. Every student and some teacher laughed joyfully.
b. Every student laughed joyfully and some teacher laughed joyfully.

(28) a. Either John or some teacher took your car.
b. Either John took your car or some teacher took your car.

Similarly the pointwise definitions mapping P_2 denotations to P_1 denotations predict, correctly, the following equivalences:

(29) a. John interviewed every bystander and a couple of storeowners.

b. John interviewed every bystander and interviewed a couple of storeowners.

(30) a. He wrote a novel or a play.
b. He wrote a novel or wrote a play.

In fact for essentially all slash categories, conjunctions and disjunctions behave pointwise. So without detailed justification we note the following equivalences.

(31) Det: most but not all students ≡ most students but not all students

(32) $(\mathrm{P}_1\backslash\mathrm{P}_1)$: He spoke softly and quickly. ≡ He spoke softly and spoke quickly.

(33) P: He lives in or near NY City. ≡ He lives in NY City or near NY City.

8.2.3 Revisiting the Coordination Generalization

We pursued our semantic analysis of coordinate expressions by interpreting a conjunction of expressions as the glb of the denotations of its conjuncts, and a disjunction as the lub of the denotation of its disjuncts. This has led us naturally towards a system in which at least certain types of expressions, boolean compounds, are directly interpreted, as we have illustrated above. Thus we independently derive and interpret (23a) and (23b) and then prove that they are logically equivalent, always denoting the same truth value.

But early work in generative grammar suggested a more syntactic approach to these equivalences. The idea was that there is only one *and* (*or*, *nor*), the S or "propositional" level one. It just combines with Ss to form Ss. Apparent coordinations of non-Ss are treated as Ss, "syntactically reduced" and *and*, *or*, and *nor* are still interpreted propositionally. So the P_1 coordination in (23a) would be derived by some Conjunction Reduction rules from the S (23b) and it would receive the same interpretation as (23b).

This syntactic approach to the initial Question thus states that what the different uses of a boolean connective have in common is that they all denote the meaning they have when they conjoin Ss. Initially this solution seems semantically appealing, since (23a) and (23b) are logically equivalent. So the reduction rules seem to satisfy Compositionality: the interpretation of the derived expression (23a) is a function (the identity function) of the one it is derived from, (23b).

But as we have seen in (21) and Exercise 8.9, this equivalence fails for most DP subjects. The relevant pairs in (34) for example are certainly not logically equivalent:

(34) a. Some student both laughed and cried.
a'. Some student laughed and some student cried.
b. Most of the students both laughed and cried.
b'. Most of the students laughed and most of the students cried.

If just one student laughed and just one, a different one, cried, (34b) is true and (34a) is false. Similarly replacing *some student* everywhere by *no student*, *exactly four students*, *more than four students*, ... and infinitely many other DPs yields sentence pairs that are not logically equivalent, though a few cases do work: *every student*, and *both Mary and Sue* preserve logical equivalence in (34) (but not if *and* is replaced by *or*).

Thus Ss derived by Conjunction Reduction are not regularly related semantically to their sources: sometimes the pairs are logically equivalent, sometimes one entails the other but not conversely, and sometimes they are *logically independent* (neither entails the other). In addition the precise formulation of the Reduction rules has not been worked out and it seems quite complicated. Note that a sentence may contain many Boolean compounds:

(35) Neither did most of the teachers write a novel or two poems or review at least one book and four plays in or near NY City nor did most of the grad students (write a novel or two poems or review at least one book and four plays in or near NY City) over the vacation.

Exercise 8.11. From our Coord rule above it follows that for every $n > 0$ there is a DP in English with more than n constituents of category DP. Prove this by induction on n and conclude that English has infinitely many Boolean compounds of DPs.

For all these reasons then we recommend directly deriving and interpreting Boolean compounds rather than deriving all from sentential sources where the Boolean connectives would be interpreted.

8.3 Negation and Additional Properties of Natural Language Lattices

The lattices we use as denotation sets have three further properties: they are *bounded*, *distributive*, and *complemented*. *not* and *neither...nor...* are interpreted by complements, which presupposes boundedness and distributivity.

Definition 8.4. A lattice $(L, \leq)$ is *bounded* iff it has a least element and a greatest element. $x \in L$ is *least* iff for all $y \in L, x \leq y$; x is *greatest* iff for all $y \in L, y \leq x$.

Fact Let $(L, \leq)$ be a bounded lattice. Then it has just one least element noted 0 read *the zero* (*bottom*), and just one greatest element, noted 1 read *the unit* (or *top*). (If x and x' are both least then $x \leq x'$ and $x' \leq x$ so by antisymmetry $x = x'$. If $(L, \leq)$ is bounded then $1 = \bigvee L$ and $0 = \bigwedge L$. Every finite lattice is bounded, since if L is finite with n elements say $a_1, \ldots, a_n$ then $\bigwedge L = (a_1 \wedge a_2 \wedge \cdots \wedge a_n)$ and similarly $\bigvee L = (a_1 \vee a_2 \vee \cdots \vee a_n)$. Most of the lattices exhibited so far are finite and thus bounded. But many non-finite lattices are bounded.

Theorem 8.5.

a. *$(\mathcal{P}(A), \subseteq)$ is bounded with A greatest and $\emptyset$ least, no matter how large A is.*
b. *If $(L, \leq)$ is bounded then the pointwise lattice $[E \to L]$ is bounded. The 0 function maps each $x \in E$ to 0_L, the zero of L, and the unit function maps each $x \in E$ to 1_L.*

Proposition 8.6. *In any lattice $(L, \leq)$, $x \vee 0 = x$.*

Proof. Since $x \leq x$ and $0 \leq x$ we have that x is an ub for $\{x, 0\}$. And for z an ub for $\{x, 0\}$, $x \leq z$, so x is least of the ub's, as was to be shown. □

Exercise 8.12. Show in analogy to the fact above that in any lattice, $x \wedge 1 = x$.

Definition 8.5. A lattice $(L, \leq)$ is *distributive* iff for all $x, y, z \in L$, (a) and (b) hold:

a. $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$, and
b. $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.

One proves in any lattice that (a) and (b) are equivalent in the sense that if either holds the other does. Moreover the right hand side of (a) stands in the $\leq$ relation to its left hand side in any lattice. So to prove that a lattice is distributive it suffices to show that $x \wedge (y \vee z) \leq (x \wedge y) \vee (x \wedge z)$, all $x, y, z \in L$. Dually, the left hand side of (b) is $\leq$ the right hand side in any lattice, so to prove a lattice distributive it suffices to show that $(x \vee y) \wedge (x \vee z) \leq x \vee (y \wedge z)$.

Theorem 8.7.

a. *The $\{T, F\}$ lattice is distributive.*
b. *All power set lattices are distributive.*
c. *If $(L, \leq)$ is distributive then so is the pointwise lattice $[E \to L]$.*

An example of a non-distributive lattice consider (15c), the pentagon lattice. There $b \wedge (d \vee c) = b \wedge a = b \neq (b \wedge d) \vee (b \wedge c) = d \vee e = d$, so distributivity fails.

Exercise 8.13. Show that the diamond lattice, (15a) fails to be distributive.

Definition 8.6. A lattice $(L, \leq)$ is *complemented* iff $(L, \leq)$ is bounded and for every $x \in L$ there is a $y \in L$ such that $(x \wedge y) = 0$ and $(x \vee y) = 1$.

Notation To say that complements are unique in a lattice $(L, \leq)$ is to say that for every $x \in L$ there is exactly one $y \in L$ satisfying the complement axioms $(x \wedge y) = 0$ and $(x \vee y) = 1$. In such a case this unique y is noted $\neg x$, read as "complement x".

Theorem 8.8. *If a complemented lattice is distributive then complements are unique.*

Proof. Suppose that $(x \wedge y) = 0$ and $(x \vee y) = 1$, and also that $(x \wedge z) = 0$ and $(x \vee z) = 1$.
Show: $y = z$.

1. $y = y \wedge 1 = y \wedge (x \vee z) = (y \wedge x) \vee (y \wedge z) = 0 \vee (y \wedge z) = y \wedge z$, whence $y \leq z$.
2. $z = z \wedge 1 = z \wedge (x \vee y) = (z \wedge x) \vee (z \wedge y) = 0 \vee (z \wedge y) = z \wedge y$, whence $z \leq y$, so by antisymmetry $y = z$.

□

Note that the third equality in each line uses distributivity. And since each x has at least one complement (y as above) we now know each x has exactly one, which we may denote $\neg x$. Here is a simple proof of the double complements law using uniqueness of complements:

Theorem 8.9. *For all x in a boolean lattice, $x = \neg(\neg x)$.*

Proof. By commutativity and axiom we have: $\neg x \wedge x = 0$ and $\neg x \vee x = 1$, Hence by uniqueness of complements, $x = \neg(\neg x)$. □

Exercise 8.14. A lattice may be complemented but not uniquely complemented (and so not distributive by Theorem 8.8). For example the pentagon lattice (15c) is complemented but the element c has two complements. One is b, since $b \wedge c = 0$ (the zero element is e) and $b \vee c = 1$, the unit element a. What is the other complement of c?

Definition 8.7. A *boolean* lattice is a lattice that is distributive and complemented.

All $\text{DEN}_{\mathcal{M}}(C)$ for C coordinable are boolean with negation denoting the complement operation, just as *and* and *or* denote the meet and join functions. The denotation for *neither...nor...* is sometimes noted $\downarrow$, defined by $x \downarrow y =_{df} (\neg x \wedge \neg y) = \neg(x \vee y)$. So *neither John nor Bill* denotes the complement of the denotation of *either John or Bill*. We now assume Eng enriched with a Negation rule:

(36) Negation: $(\text{not}, \text{NEG}) + (s, C) \to (\text{not } s, C)$

Fact

1. In the $\{T, F\}$ lattice, T is the top or unit element, F is the bottom or zero element, and provably $\neg T = F$ and $\neg F = T$.
2. In any power set lattice $\mathcal{P}(A)$, for each $X \subseteq A$, $\neg X$ is provably $A - X$, the set of elements in A that are not in X.
3. In a pointwise boolean lattice $[E \to L]$, $\neg F$ provably maps each $x \in E$ to $\neg(F(x))$. For example interpreting *not* as the complement operator in the pointwise $[\mathrm{P}_1 \to \mathrm{P}_0]$ lattice we have:

(37)
$$\begin{array}{cccc} \text{not} & \text{every} & \text{student} & \text{laughed} \\ | & | & | & | \\ [\![\cdot]\!] & [\![\cdot]\!] & [\![\cdot]\!] & [\![\cdot]\!] \\ | & | & | & | \\ \neg & \text{EVERY} & \text{STUDENT} & \text{LAUGH} \end{array}$$

$$\text{EVERY}(\text{STUDENT})$$
$$\neg(\text{EVERY}(\text{STUDENT}))$$
$$(\neg(\text{EVERY}(\text{STUDENT})))(\text{LAUGH})$$
$$= \neg((\text{EVERY}(\text{STUDENT}))(\text{LAUGH}))$$

"Not every student laughed."

Remark In each lattice L above, the complement operation is not simply added to L, rather, the partial order provably is a bounded distributive lattice with the property that for each x there is a unique y such that $x \wedge y = 0$ and $x \vee y = 1$. Then we *define* $\neg x$ to be that unique y. For example in the $\{T, F\}$ lattice with the implication order, F is provably least and T greatest, and $T \wedge F = F$, the zero, and $T \vee F = T$, the unit, so we infer that F is the complement of T, that is, $F = \neg T$.

Interpreting negation as complement generally yields reasonable results in terms of judgment of entailment and logical equivalence. And it answers the query analogous to the one we raised for *and* and *or*. Namely, the uses of negation with expressions in different categories do have something in common: they always denote the boolean complement of the denotation of the expression they negate. But they appear to combine with somewhat fewer categories and *and* and *or* and to exhibit more category internal restriction.

The most easily negated expressions across languages are P_1s (despite a tradition that calls it "sentential" negation). In English the expression of this negation is fully natural, but complicated. It requires the presence of an auxiliary verb, an appropriately tensed form of *do*, as in (38b).

(38) a. Just two skaters fell.
b. Just two skaters didn't fall.
c. It is not the case that just two skaters fell.

Note that (38c) is not at all logically equivalent to (38b). In a situation with exactly four skaters, just two of whom fell both (38a) and (38b) are true, and (38c) is false, so (38b) does not entail (38c). And in a situation with six skaters, exactly three of whom fell, (38c) is true and (38b) false, so (38c) fails to entail (38b). The point of this observation is that the information contained in the subject of the P_1 is not in general understood to be under the scope of P_1 negation.

Equally many DPs negate easily, as in (39), but also many don't, as in (40).

(39) a. Not a creature was stirring, not even a mouse.
b. Not more than a couple of students will answer that question correctly.
c. Not one student in ten knows the answer to that.
d. Not every student came to the party.

(40) a. *Not John came to the party.
b. *Not the students I met signed my petition.
c. *Not each student came to the party.

On the other hand sometimes apparently unnegatable DPs can be forced to negate in coordinate contexts, as in *So Sue and not Jill will represent us at the meeting.*

Finally we note that it is usually quite difficult to interpret negation as taking a mere P_2 (or P_3) in its scope. *John didn't criticize every teacher* does not mean that John stands in the not-criticize relation to every teacher, which would mean that every teacher has the property that John didn't criticize him. Rather the sentence most naturally means simply that John lacks the property expressed by *criticized every* *teacher*.

8.4 Properties versus Sets: Lattice Isomorphisms

We have already said (Chapter 3) what it means for two relational structures, in particular two (boolean) lattices, to be isomorphic: you must be able to match the elements of the their domains one for one in such a way that elements stand in the order relation in one if and only if their images stand in the order relation in the other. Now for the lattices we have considered there is one interesting and possibly not obvious, instance of an isomorphism that is used often in the literature, often without explicit mention. Namely, a power set lattice $(\mathcal{P}(A), \subseteq)$ is isomorphic to the corresponding pointwise "property" lattice $([A \rightarrow \{T, F\}], \leq)$. To show this we exhibit an isomorphism. Let K be any subset of A and define h_K from A into $\{T, F\}$ by

(41) $h_K(a) = T$ iff $a \in K$.

Now we claim that the function h mapping each subset K of A to h_K is an isomorphism. Here is an informal proof using the Hasse diagrams of the lattices. First the power set lattice, repeating (16).

(42)
$$\begin{tikzcd}[arrows=dash]
& \{a,b,c\} \arrow[dl] \arrow[d] \arrow[dr] & \\
\{a,b\} \arrow[d] \arrow[dr] & \{a,c\} \arrow[dl] \arrow[dr] & \{b,c\} \arrow[dl] \arrow[d] \\
\{a\} \arrow[dr] & \{b\} \arrow[d] & \{c\} \arrow[dl] \\
& \emptyset &
\end{tikzcd}$$

And now consider the Hasse diagram for the h_K:

(43)
$$\begin{tikzcd}[arrows=dash]
& h_{\{a,b,c\}} \arrow[dl] \arrow[d] \arrow[dr] & \\
h_{\{a,b\}} \arrow[d] \arrow[dr] & h_{\{a,c\}} \arrow[dl] \arrow[dr] & h_{\{b,c\}} \arrow[dl] \arrow[d] \\
h_{\{a\}} \arrow[dr] & h_{\{b\}} \arrow[d] & h_{\{c\}} \arrow[dl] \\
& h_{\emptyset} &
\end{tikzcd}$$

Now it is clear that the map h sending each K in (42) to h_K in (43) is a bijection. But two queries still arise. First, how do we know that all the maps from A into 2 (we often write 2 for $\{T, F\}$ recall) are exhibited in (43)? The answer is easy: for any g from A into 2 let $T[g]$

be $\{a \in A \mid g(a) = T\}$, the set of elements of A that g is true of. Then clearly $h_{T[g]}$ is g, since $h_{T[g]}$ is true of exactly the elements of $T[g]$, so $h_{T[g]}$ and g are true of exactly the same objects. And second, how do we know that we have correctly represented the $\leq$ relation in (43)? Well, we see that in moving up along lines from some h_K to some $h_{K'}$ it must be so that $K \subseteq K'$, whence the set of things h_K maps to T is a subset of those that $h_{K'}$ maps to T; that is, for all $a \in A$, if $h_K(a) = T$ then $h_{K'}(a) = T$, and this suffices to show that $h_K \leq h_{K'}$, completing the proof.

Now, given that a power set lattice and its pointwise counterpart are isomorphic, why should we care? One practical reason is that authors differ with regard to how they represent properties of objects. Often we find it natural to think of a property of objects X as a function that looks at each element of X and says True (or False). So we treat the set of properties of objects X as $[X \to \{T, F\}]$. And in such a case when b is an object and p a property we write $p(b)$ to say that the object b has property p. But other times we just treat a property of elements of X as the set of objects which have it. So here the set of properties is $\mathcal{P}(X)$, and we write $b \in p$ to say that b has property p.

But from what we have just seen, $P(X)$ and $[X \to \{T, F\}]$ are isomorphic, and we write $P(X) \approx [X \to \{T, F\}]$. That means that the order theoretic claims we can make truly of one are exactly the ones we can make about the other. Whenever one says $b \in p$ the other says $p(b)$ and conversely. In fact within a given text an author may shift back and forth between notations, acknowledging that there is no logical point in distinguishing between isomorphic structures.

We close with two further properties which the boolean lattices we use have but which are not present in all boolean lattices.

Definition 8.8. A lattice $(L, \leq)$ is *complete* iff every subset has a glb and a lub, that is, $\bigwedge K$ and $\bigvee K$ exist for all $K \subseteq L$.

Fact All finite lattices are complete. If $K = \{k_1, \ldots, k_n\} \subseteq L$ then $\bigwedge K = k_1 \wedge \cdots \wedge k_n$ and $\bigvee K = k_1 \vee \cdots \vee k_n$. $\bigwedge L$ is the zero element of L and $\bigvee L$ is the unit.

Definition 8.9.

a. An element α of a lattice is an *atom* iff $\alpha \neq 0$ and for all x, if $x \leq \alpha$ then $x = 0$ or $x = \alpha$

b. A lattice $(B, \leq)$ is *atomic* iff for all $y \neq 0$ there is an atom $\alpha \leq y$. Write ATOM(B) for the set of atoms of B.

In a power set boolean lattice the unit sets are the atoms.

Theorem 8.10. *A boolean lattice* $(B, \leq)$ *is complete and atomic iff it is isomorphic* $\mathcal{P}(ATOM(B))$.

The map sending each y to $\{\alpha \in \text{ATOM}(B) \mid \alpha \leq y\}$ is the desired isomorphism. Thus $|B| = |\mathcal{P}(\text{ATOM}(B))| = 2^{|\text{ATOM}(B)|}$.

Theorem 8.11. *All finite boolean lattices are complete and atomic. So each finite boolean lattice is isomorphic to the power set of its atoms, and so isomorphic to a power set.*

There are complete atomic distributive lattices that are not boolean, as they are not complemented. Here is one that arises in certain semantic studies.

(44) Let A be a non-empty set and $EQ(A)$ the set of equivalence relations on A. So $R \in EQ(A)$ iff $R \subseteq A \times A$ and R is reflexive, symmetric and transitive. Define a binary relation, called *refines*, on $EQ(A)$ by setting $R \leq S$ iff for all $x, y \in A$, $xRy \Rightarrow xSy$. So whenever s is R-equivalent to y then it is also S-equivalent to y.

The above definition implies that every S-equivalence class is a union of R-equivalence classes. Furthermore, the definition of $\leq$ basically says that $R \subseteq S$, so we know then that $\leq$ is reflexive, antisymmetric and transitive.

Fact $\langle EQ(A), \leq \rangle$ is a poset.

Further, since each equivalence relation over A is reflexive we have that $ID_A = \{\langle x, x \rangle \mid x \in A\}$ is a subset of every R in $EQ(A)$. And since ID_A is itself an equivalence relation (antisymmetry and transitivity are vacuously satisfied) we have that ID_A is least. That is, the 0 of $EQ(A)$ is ID_A. Similarly $A \times A$ is the unit 1 of $EQ(A)$ since it is an equivalence relation on A and all (equivalence) relations on A are subsets of $A \times A$. So $EQ(A)$ is a bounded poset. What is slightly less obvious is that $EQ(A)$ is a lattice. Glbs are actually intuitive since:

Theorem 8.12.

a. *For all* $R, S \in EQ(A)$ *we have that* $R \cap S$ *is an equivalence relation in* $EQ(A)$. *Generalizing,*

b. *For* I *an index set, if* $R_i \in EQ(A)$ *for each* $i \in I$, *then* $\bigcap_i R_i$ *is an equivalence relation (so* $\bigcap_i R_i \in EQ(A)$*).*

Proof. We prove the general case by showing that $\bigcap_i R_i$ is reflexive, symmetric, and transitive.

Reflexivity. Let $x \in A$ be arbitrary. Then for each I, $\langle x, x\rangle \in R_i$ so it is in their intersection, thus $\bigcap_i R_i$ is reflexive.

Symmetry. Let $\langle x, y\rangle \in \bigcap_i R_i$. So $\langle x, y\rangle$ is in each R_i. Since each R_i is symmetric, $\langle y, x\rangle$ is in each R_i, so it is in $\bigcap_i R_i$ showing that $\bigcap_i R_i$ is symmetric.

Transitivity. Similarly if $\langle x, y\rangle$ and $\langle y, z\rangle$ are both in $\bigcap_i R_i$ then they are both in each R_i so $\langle x, z\rangle$ is in each R_i since each R_i transitive. Thus $\langle x, z\rangle \in \bigcap_i R_i$, showing that the latter is transitive. Thus $\bigcap_i R_i$ is an equivalence relation. We already know it is glb for $\{R \mid i \in I\}$. □

The trickier part is to show that any collection of equivalence relations on A has a least upper bound. That lub would be the union of the collection if that union were an equivalence relation, but often isn't. For example, let x, y, z be distinct element of A. Then,

(45) a. $R = ID_A \cup \{\langle x, y\rangle, \langle y, x\rangle\}$ is an equivalence relation over A.
b. $S = ID_A \cup \{\langle y, z\rangle, \langle z, y\rangle\}$ is an equivalence relation over A.
c. $R \cup S$ is not an equivalence relation as it contains $\langle x, y\rangle$ and $\langle y, z\rangle$ but not $\langle x, z\rangle$ so it fails transitivity.

But in fact any collection of equivalence relations over A has a lub in $EQ(A)$, but that relation is larger than the union of the R_i in the collection. Here is a standard way to build it.

Definition 8.10. Given an index set I with $R_i \in EQ(A)$ for each $i \in I$, let K be the following set:

$$\bigcap\{S \in EQ(A) \mid \text{for all } i \in I, R_i \subseteq S\}.$$

Clearly K is an equivalence relation over A by the proof of Theorem 8.12. It is obviously the least equivalence relation that includes each R_i (meaning, as usual, that it is a subset of each equivalence relation over A which includes each R_i; this is so because the intersection of a bunch of sets is always a subset of any set in over which the intersection was taken. So K is the least upper bound for $\{R_i \mid i \in I\}$. Thus $(EQ(A), \leq)$ is a complete bounded lattice.

$(EQ(A), \leq)$ is not however a boolean lattice, as in general its elements lack complements. For example, for A with at least three distinct elements x, y, z the relation R in 45 lacks a complement in $EQ(A)$. If Some R^* were its complement then its intersection with R must be ID_A, so R^* must lack $\langle x, y\rangle$ and $\langle y, x\rangle$. If it lacks any other pair then its union with R will not be $A \times A$. So it must then have the pairs $\langle x, z\rangle$ and $\langle z, y\rangle$, whence by transitivity it has $\langle x, y\rangle$, a contradiction.

Thus we have shown that $(EQ(A), \leq)$ is a complete lattice which in general is not boolean. In fact more can be said:

Exercise 8.15.

a. Show that $(EQ(A), \leq)$ is atomic and exhibit an atom.
b. Show that $(EQ(A), \leq)$ is distributive.

8.5 Theorems on Boolean Lattices

We conclude this chapter with some basic regularities that hold in all boolean lattices. For those that are named, the names are in common use and should be learned.

Theorem 8.13. *For $(B, \leq)$ a boolean lattice and any x, y, z in B,*

a. $x \wedge y = y \wedge x$, and $x \vee y = y \vee x$. (Commutativity)
b. $x \wedge (y \wedge z) = (x \wedge y) \wedge z$, and $x \vee (y \vee z) = (x \vee y)$. (Associativity)
c. $x \wedge x = x$, and $x \vee x = x$. (Idempotency)
d. $\neg\neg x = x$. (Double Complements)
e. $\neg(x \wedge y) = \neg x \vee \neg y$, and $\neg(x \vee y) = \neg x \wedge \neg y$. (DeMorgan's Laws)
f. $x \wedge (x \vee y) = x$, and $x \vee (x \wedge y) = x$. (Absorption)
g. $x \leq y$ iff $(x \wedge y) = x$, and $x \leq y$ iff $(x \vee y) = y$.
h. $x \leq y$ iff $x \wedge \neg y = 0$, and $x \leq y$ iff $\neg x \vee y = 1$.
i. $x \wedge y \leq x \leq x \vee y$.
j. $x \leq y$ iff $\neg y \leq \neg x$.
k. $x \leq y \rightarrow (x \wedge z) \leq (y \wedge z)$, and $x \leq y \rightarrow (x \vee z) \leq (y \vee z)$.

Exercise 8.16. Let (A, R) be a relational structure. Define a binary relation R^{-1} on A by setting $xR^{-1}y$ iff yRx. R^{-1} is called the *converse* of R. We often use symbols like $\leq$, $\preceq$, $\subseteq$ for partial orders, and $\geq$, $\succeq$, $\supseteq$ for their converses.

a. Prove that when (A, R) is a poset then (A, R^{-1}) is a poset. (A, R^{-1}) is called the *dual* of (A, R).
b. Same as (a) above with 'lattice' replacing 'poset'.
c. Given a boolean lattice (A, R) prove that the complement function from A to A is an isomorphism from (A, R) to its dual (A, R^{-1}).

8.6 A Concluding Note on Point of View

We have presented lattices in an order theoretic way: $x \wedge y$ and $x \vee y$ are defined as greatest lower bounds and least upper bounds. Another widely used approach, perhaps the most widely used, is one in which a lattice is given as a triple $(L, \wedge, \vee)$, with $\wedge$ (meet) and $\vee$ (join) binary functions on L satisfying commutativity, associativity, and absorption.

Then we define $\leq$ by: $x \leq y$ iff $x \wedge y = x$. Sets with functions defined on them are algebras, so on this view a lattice is an algebra of a certain sort, and a boolean lattice is called a boolean algebra, named after George Boole (1854) who first constructed them. The two approaches, the relational one and the functional one, are interdefinable and serve to illustrate different ways of accomplishing the same goal. Recall,

If you can't say something two ways you can't say it.

Boole's speculation. Boolean algebra has developed explosively since Boole initiated it. But Boole's original work still merits reading, especially for its motivation. Boole was not interested in inventing a type of algebra per se, rather he was trying to formulate with mathematical precision and rigor the thought steps he took in clear reasoning, hence his title *The Laws of Thought*. This was a marvelously ambitious enterprise, and while we may reasonably think there is more to thought than the kinds of reasoning that can be carried out with in boolean algebra, might not Boole's intuitions give us a deeper account of the polymorphism of *and*, *or*, and *not*? Their meanings indeed are not tied to any particular type of denotation truth value, property, relation, restricting modifiers, generalized quantifiers, ... and this suggests that the boolean operators express *properties of mind*, more the way we think about things, how we conceptualize them, than properties of things themselves. This suggestion is to be sure speculative.

But there is one additional linguistic observation that is nicely compatible with it. Namely, if the Boolean connectives have a meaning independent of the categories of expression they combine with, as we have supported here, then it is perhaps not surprising that they have some capacity to create meaningful constituents for which there is little or no independent syntactic support. We have already noted cases like (46a,b):

(46) a. *So not Jill will represent us at the meeting.
b. So Sue and not Jill will represent us at the meeting.

Similarly in (47a) the right branching constituency of the subject DP is natural, but a left branching constituency can be forced by coordination.

(47) a. [Most [female [doctors]]] support healthcare reform.
b. [[Most female] and [almost all male]](doctors) support healthcare reform.

Right node raising cases like (48a,b) and Gapping as in (49a,b) are further instances:

(48) a. [John [bought a turkey]] and [Bill [cooked it]].
b. [[John bought] and [Bill cooked]] the turkey.

(49) a. John [[handed Bill] a snake].
b. John [handed [Bill a snake] and [Fred a scorpion]].

Steedman and Baldridge (2007) provide derivations of various "non-constituent coordinations" within the framework of Combinatory Categorial Grammar. Here we offer no such derivational mechanisms, we simply notice that however such coordination and negation is derived, it is the Boolean connectives that induce or allow it.

8.7 Further Reading

See Payne (1985) and Horn (1989) for typological discussion of negation; see Keenan and Faltz (1985) for extensive discussion of boolean structure in natural language semantics. And see Winter (2001) for an analysis of the interaction of Boolean structure with plurals and collectives, as well as issues concerning wide scope phenomena (which we consider in Chapter 9).

9

Semantics III: Logic and Variable Binding Operators

Generalized quantifiers and boolean lattices are powerful conceptual tools for representing aspects of natural language semantics. We pursue them more extensively in Chapters 10 and 11. Here we address two independent semantically significant topics that are enlighteningly treated within standard logic and its extensions. These concern scope ambiguities and "binding" phenomena. The tool we use from standard logic is *variable binding operators* (VBO's).

This chapter is presented in three parts: Section 9.1 contains the syntax and semantics of standard first order logic (FOL), together with basic examples of its utility in "translating" expressions from ordinary English. Section 9.2 steps back and summarizes a variety of general linguistic properties of FOL. The idea is to give the reader a feel for the "linguistic" character of work in logic, as well as to present several of the major results logicians have achieved. This section is intended for reference and is not presupposed by the subsequent sections and chapters of this book. Finally, Section 9.3 Presents the syntax and semantics of the lambda operator, an enormously useful conceptual tool in the representation of semantic properties of natural language.

9.1 Translation Semantics

It is only recently[1] that the direct interpretation of natural language expressions has become feasible. More traditionally the semantic analysis of natural language proceeded by translation into logic, usually some variant of FOL. Such an approach is helpful because FOL is well understood, so translating English say into a first order language has

[1] The direct interpretation tradition can be said to have begun with Montague (1974).

the merit of representing something we are trying to understand in terms of something we already understand.

9.1.1 Semantic phenomena motivating the use of variable binding operators

Scope ambiguities

Scope ambiguities have already been seen in simple examples like (1), which can be understood on the *object narrow scope* (ONS) reading in (1a), a reading our direct interpretation analysis already captures, and on the *object wide scope* (OWS) reading in (1b), which our analysis to date does not capture.

(1) Some student praised every teacher.

a. ONS: There is at least one student who praised every teacher.

b. OWS: For every teacher there is a student who praised him (possibly different teachers were praised by different students).

So on the OWS reading the students may vary with the teachers. But on the ONS reading a student is chosen independently of the teachers and it is asserted that that student stands in the PRAISE relation to each teacher. (It is not ruled out that there is more than one student with this property). It is easy to image a situation in which (1) is true understood on the OWS reading but false on the ONS reading. Imagine a situation with just two students, John and Mary, and just two teachers. John praises one of the teachers and no one else, and Mary praises the other teacher and no one else. Then for each teacher there is a student who praised him, so (1b), the OWS reading of (1), is true. But no one student praised each teacher, so (1a), the ONS reading, is false.

In mathematical discourse scope ambiguities would be intolerable, whence the utility of variable binding notation there. Compare the strikingly different meanings of the two scope readings of (2) for example, into which we introduce an informal use of variables.

(2) Some number is greater than every number.

a. ONS: There is a number x such that for every number y, $x > y$.

b. OWS: for every number y there is a number x such that $x > y$.

In elementary arithmetic (2b) is true: given y, choose x to be $y + 1$. But (2a) is false, a number greater than every number would be greater than itself, an impossibility.

Our concern in this chapter is with scope ambiguities involving pairs of DPs as illustrated here. But scope ambiguities arise for other expression types as well. For example (3) is ambiguous according to the scope of negation:

(3) John didn't leave because the children were crying.
 a. John stayed, because the children were crying.
 b. John left, but not because the children were crying.

On the reading in (3a) the subordinate clause, *because the children were crying*, is not in the scope of the negation. Rather that clause modifies *didn't leave* and the sentence means roughly *Because the children were crying John didn't leave*. In (3b) the subordinate clause modifies *leave* and the entire modified predicate is negated, so the subordinate clause in particular is in the scope of *didn't*. (3b) doesn't deny that *John left*, it only denies that the children crying was the reason for his leaving.

Binding

There is an ambiguity in (4), not entirely dissimilar to the scope ambiguity in (2):

(4) Each of the children loves his mother.

On one reading *his* is understood as *bound* to *each of the children* and (4) would be true in a situation with several children as long as each one loves his own mother—Johnny loves his mother, Amy loves her mother, etc. So on this reading the mothers may vary with the children.

But (4) has a second reading, less apparent than the first when, as here, no context is provided. On this reading *his mother* refers to some woman whose existence has been previously established in the discourse. Imagine that we have been discussing Billy, whose mother is the local kindergarten teacher beloved by all the children. Then an assertion of (4) might be used to assert that each child in the discourse loves Billy's mother. So on this reading context provides a denotation for *his (mother)*, and the choice of mothers does not vary with the choice of children.

Compare the use of *his* in (4) with that of *its* in the more mathematical (5).

(5) Every number is less than or equal to its square.
 a. For every number x, x is less than or equal to x squared.
 b. For every number x, x is less than or equal to y squared.

Out of context the natural reading of (5) is (5a), in which the pronoun *its* is bound to *every number*. On this reading (5) implies that 3 is less than or equal to 3 squared (usually written $3 \leq 3^2$), $13 \leq 13^2$, and in general, for each number n, $n \leq n^2$.

Notice that the portion of (5a) following the quantifier phrase *for every number* x, uses the same variable x twice. This is what tells us that we compare 3 with 3^2, 13 with 13^2, etc. The two occurrences of x are said to both be *bound* by the quantifier phrase *for every number* x. In contrast, in (5b) we have used different variables in the portion following the quantifier phrase. Here the first occurrence, x, is bound by the quantifier phrase since its variable matches x. But the second variable y is not bound by the quantifier phrase. This reading of (5) is comparable to the non-bound reading of *his mother* in (4), where context identifies a referent for the phrase and it does not vary with the choice of child. In (5) y denotes some number fixed in context, but (5) itself provides no context to help us figure out the denotation of y.

This completes our illustration of the phenomena we are to represent. We turn now to an informal presentation of first order logic.

9.1.2 First Order Logic

First Order Logic (FOL) defines a class of languages, called *first order languages*, and states how expressions in each of these languages is semantically interpreted. So FOL is a kind of universal grammar. Different first order languages differ by their choices of lexical items, specifically their n-place predicate symbols and n-place function symbols (individual constants are 0-place function symbols). For example, in the language of Set Theory we would take a single two place predicate symbol, $\in$, as primitive (and then of course state many axioms that use the predicate); in the language of Elementary Arithmetic we might have two two place function symbols, $+$ and $\cdot$, as primitive; in Euclidean geometry we would have a two place predicate symbol $\|$ read as *is parallel to* and a three place predicate symbol B read as *is between*. In linguistic parlance these are the *parameters* of the language, and are called *non-logical constants*. Once given, all first order languages form complex expressions in the same ways (below).

Syntax:

Terms are expressions which, when interpreted, denote elements in the domain $E_{\mathcal{M}}$ of a model $\mathcal{M}$ (defined shortly). Syntactically unanalyzable terms are either *individual variables*: $x, y, z, x_1, y_1, \ldots$, of which there are always denumerably many (= natural number many), or *individual constants* such as '0', '1', etc., of which there are usually just a few, sometimes none. Then complex terms are built recursively by combining function symbols, such as '+'

in the language of arithmetic, with an appropriate number of terms to form a complex term. We usually write two place function symbols between their arguments, writing $(0 + 1)$ in preference to $+(0, 1)$. Since we use infix notation some parentheses are needed to rule out pernicious ambiguities. For example, $2 \cdot (3 + 4) = 14 \neq (2 \cdot 3) + 4 = 10$ so if we eliminated all parentheses we would have ambiguously denoting expressions (as we saw earlier in discussing the placement of boolean connectives).

Formulas are the sort of expression which we think of as True or False under an interpretation, and they come in three syntactic types: *Atomic Formulas*, *Boolean Compounds*, and *Quantified Formulas*.

Atomic Formulas are built by combining an n-place predicate symbol P with n terms, $t_1, \ldots, t_n$, the result being noted $P(t_1, \ldots, t_n)$, which is called an *atomic* formula (even when the terms themselves are syntactically complex). Often when P is a two place predicate symbol we write $(t_1\, P\, t_2)$ rather than $P(t_1, t_2)$. So usually we write $(0 \leq 1)$ rather than $\leq (0, 1)$, so once again parentheses are necessary. Note that the standard syntax for first order languages treats P_2s as combining directly with pairs of expressions to form a P_0 (formula) rather than combining with just one to form a P_1, which in turn combines with one term to form a P_0. In the pair notation the "subject" argument is written first. So using English expressions, *Kim praised Amy* would be represented as *praise(Kim, Amy)*, whereas we have been writing *((praise Amy) Kim)*, with *praise Amy* a constituent of category P_1.

Boolean Compounds of formulas are formulas. Here we write '&' for 'and', 'or' for 'or', $-$ for 'not', $\rightarrow$ for 'if...then...' and $\leftrightarrow$ for 'if and only if'. Different authors use different variants of this notation. So if P and Q are formulas so are $(P \,\&\, Q)$, $(P$ or $Q)$, $-P$, $(P \rightarrow Q)$ and $(P \leftrightarrow Q)$. First order languages don't allow direct boolean compounds of P_1s (or P_ns for any $n > 0$), though expressive power would not be changed if we added this in.

Quantified Formulas are ones formed by concatenating a quantifier symbol followed by a variable, followed by a formula. There are two quantifier symbols: $\forall$, the universal quantifier symbol, read as "for all", and the existential quantifier symbol $\exists$, read as "for some", or "there exists". For example, in the language of elementary arithmetic $\forall x(x < x + 1)$ is a universally quantified formula read as "For all numbers x, x is less than x plus one". And $\exists x(\mathit{Prime}(x) \,\&\, \mathit{Even}(x))$ is an existentially quantified formula read as "There is a number x such that x is prime and x is even". Note that there is nothing in the formulas themselves that tells us what we are using the quantifiers to quantify over (the natural numbers). It was the stipulation that these Ss were drawn from the language of elementary arithmetic that guarantees this in this case, since that language is just used to speak about the natural numbers.

Semantics (informal): Individual constants denote elements of the domain of a model. A derived term $F(t_1, \ldots, t_n)$ denotes the value of the function denoted by F at the n-tuple of objects denoted by the n terms $(t_1, \ldots, t_n)$. An atomic formula $P(t_1, \ldots, t_n)$ is true in a model $\mathcal{M}$ iff the n-tuple of objects denoted by the n-ary sequence $(t_1, \ldots, t_n)$ terms stands in the relation denoted by P. An equivalent approach treats 'P' as denoting a function mapping n-tuples of objects from the domain E into $\{T, F\}$. The symbol '$=$', a logical constant, denotes the identity relation, $\{\langle b, b\rangle \mid b \in E\}$.

Boolean compounds of formulas are interpreted as expected: A conjunction of formulas $(P \,\&\, Q)$ is interpreted as T iff each conjunct is T, a disjunction is interpreted as T iff at least one disjunct is T. A conditional $(P \to Q)$ is interpreted as T iff P is interpreted as F or both P and Q are interpreted as T. $(P \leftrightarrow Q)$ is interpreted as T iff P and Q are interpreted as the same truth value (both T or both F).

For quantified formulas, $\forall x\varphi$ is interpreted as T iff φ is interpreted as T no matter what object in the domain we let x denote. $\exists x\varphi$ is interpreted as T iff there is an object b in the domain of the model such that φ is interpreted as T when we set x to denote that b. To say this in a formally rigorous way we will need a mechanism which lets the denotations of variables vary holding constant the denotations of the other lexical items (the predicate and function symbols).

For example, in the first order language of Elementary Arithmetic the formula $\forall x(x \leq x^2)$ is true in the model whose domain is $\mathbb{N}$. No matter what $n \in \mathbb{N}$ we choose, $x \leq x^2$ is true when x is set to denote n. Similarly $\exists x(x = x^2)$ is true since letting x denote 1 the formula $x = x^2$ is true. That formula is also true when x is set to denote 0, but it is false when x denotes any $n > 1$.

Crucially in a *first order language* $\mathcal{L}$ we may have symbols denoting functions or relations (on the domain of the model), but we cannot use quantifiers with variables ranging over such functions or relations. Thus (6a) is a first order sentence. It says that the function 'h' denotes is one to one. But (6b), which says that there is a one to one function is not a first order sentence.

(6) a. $\forall x \forall y(h(x) = h(y) \rightarrow x = y)$
b. $\exists h \forall x \forall y(h(x) = h(y) \rightarrow x = y)$

Similarly (7a), which says that R is a symmetric relation, is first order, but (7b), which defines symmetry, is not:

(7) a. $\forall x \forall y(xRy \leftrightarrow yRx)$
b. $\forall R(R\ \textit{is symmetric} \leftrightarrow \forall x \forall y(xRy \leftrightarrow yRx))$

9.1.3 Representing English in First Order Logic

Representing English in FOL takes practice (and is not always possible, as we see below). The major artificiality is that DPs such as *every student*, *some student*, etc. do not occur as arguments of predicates. Consider the natural first order translations of the English Ss below:

(8) a. John criticized every student.
b. $\forall x(\text{STUDENT}(x) \rightarrow \text{CRITICIZE}(\text{JOHN}, x))$
c. For every object x, if x is a student then John criticized x.

(8b) is a standard first order rendition of (8a), though we would often abbreviate JOHN simply as 'j'. A literal read-out of (8b) is (8c), which makes it clear that we are quantifying over all objects in the domain, not just the students. Moreover no constituent of (8b) or (8c) corresponds to *every student* in (8a). Compare (8a) with (9a) below:

(9) a. John criticized some student.
b. $\exists x(\text{STUDENT}(x)\ \&\ \text{CRITICIZE}(\text{JOHN}, x))$
c. There is an object x such that x is a student and John criticized x.

Again, (9b), as made clear in (9c), quantifies over the entire domain of objects, not just the students. And the constituent *some student* in (9a) does not correspond to any constituent in (9b). Note too that both (8b) and (9b) introduce VBO's, $\forall x$ and $\exists x$ respectively, and in each case two occurrences of a variable in the following formula are bound, even though nothing in (8a) or (9a) indicates that binding is required (they contain no pronouns or DP gaps for example). Equally both (8b) and (9b) introduce a boolean operator, $\rightarrow$ in the case of (8b) and & in the case of (9b). The difference in boolean operator correlates with the difference in the choice of quantifier, $\forall$ vs. $\exists$. But that there are such differences at all is surprising. After all (8a) and (9a) appear to be syntactically isomorphic, differing just by a choice of lexical item, *every* vs. *some*.

These "unnaturalness" facts highlight that translations of English into FOL are not syntactically driven. The criterion of good translation is whether it gets the entailment properties right. Compare (8a) and

(9a) with (10a) below, also apparently isomorphic to the first two.

(10) a. John criticized no student.
b. $-\exists x(\text{STUDENT}(x)\ \&\ \text{CRITICIZE}(\text{JOHN}, x))$
b'. $\forall x(\text{STUDENT}(x) \rightarrow \text{CRITICIZE}(\text{JOHN}, x))$
c. It is not the case that for some object x, x is a student and John criticized x.
c'. For all objects x, if x is a student then it is not the case that John criticized x.

Here both (10b) and (10b') are reasonable translations of (10a) as they have the same, correct, truth conditions, hence the same entailments. And in general as the English sentences increase even slightly in complexity we find that the syntactic complexity of the FOL translations often skyrockets. Here are a few illustrative examples (shortening 'JOHN' to 'j').

(11) a. John criticized two students.
b. $\exists x \exists y(\text{STUDENT}(x)\ \&\ \text{STUDENT}(y)\ \&$
$-(x = y)\ \&\ \text{CRITICIZE}(j, x)\ \&\ \text{CRITICIZE}(j, y))$

So while (11a) appears syntactically isomorphic to (10a), its FOL translation uses two VBO's and five conjunctions. Further (11b) is true in any model in which John criticized at least two students, in particular ones in which he criticized a dozen students. To force an upper bound on the number criticized English can use little words like *only*, *exactly*, and *just* which effect only a modest increase in syntactic complexity. But consider (12a) and a reasonable FOL translation, (12b), which we may literally read as (12c).

(12) a. John criticized exactly one student.
b. $\exists x(\text{STUDENT}(x)\ \&\ \text{CRITICIZE}(j, x)$
$\&\ \forall y((\text{STUDENT}(y)\ \&\ \text{CRITICIZE}(j, y)) \rightarrow y = x))$
c. John criticized a student, and every student who John criticized is that one.

Exercise 9.1. Provide reasonable FOL translations for each of the following:

a. John criticized three students.
b. John criticized just two students.
c. John criticized a student and a teacher.
d. Every student criticized John.
e. Not every student criticized every student.

Now let us turn to some of the cases that motivate our interest in VBO's. Consider first the representation of each of the readings of (13)

(13) Some student praised every teacher.

a. ONS: There is at least one student who praised every teacher.

a'. $\exists x(\text{STUDENT}(x)\,\&\,\forall y(\text{TEACHER}(y) \rightarrow \text{PRAISE}(x,y)))$

b. OWS: For every teacher there is a student who praised him.

b'. $\forall y(\text{TEACHER}(y) \rightarrow \exists x(\text{STUDENT}(x)\,\&\,\text{PRAISED}(x,y))$

We have observed that (13b') does not entail (13a'). Here is an informal model which shows this: Let TEACHER $= \{\text{Mary}, \text{Sue}\}$, and let STUDENT $= \{\text{Manny}, \text{Moe}, \text{Jack}\}$. Lastly, let Manny PRAISE Mary, and both Moe and Jack PRAISE Sue and no one else PRAISEs anyone else. Then for every teacher y we can find a student x such that x praised y. In fact in Sue's case we can find two such x. However there is no student x in this model who stands in the PRAISE relation to both Mary and Sue, hence (13a') is false in this model. A rather more typical case for Ss with scope ambiguities is that neither scope reading entails the other.

Exercise 9.2. Exhibit FOL translations of the two scope readings for the S displayed below. Then exhibit an informal model on which the ONS is true and the OWS false, showing that ONS$\nrightarrow$ OWS. Then exhibit another model in which the OWS reading is true and the ONS one false, showing that OWS $\nrightarrow$ ONS.

Two students criticized two teachers.

Notice that coordinate P_1s in English must be translated into first order $\mathcal{L}$s with coordination at the $\text{S} = \text{P}_0$ level.

(14) a. Some student both laughed and cried.

$\exists z(\text{STUDENT}(z)\,\&\,\text{LAUGH}(z)\,\&\,\text{CRY}(z))$

b. Some student laughed and some student cried.

$\exists z(\text{STUDENT}(z)\,\&\,\text{LAUGH}(z))\,\&\,\exists x(\text{STUDENT}(x)\,\&\,\text{CRY}(x))$

Clearly (14a) entails (14b) since if x is a student who both laughed and cried then x laughed, whence some student laughed, and also x cried, whence some student cried, so (14b) is the conjunction of two true Ss and thus is true. But (14b) does not entail (14a). If John and Bill are the only students, John laughed but didn't cry and Bill cried but didn't laugh then (14b) is true and (14a) false.

Exercise 9.3. Exhibit FOL translations of (a) and (b). Say why (b) entails (a) and exhibit an informal model which shows that (a) does not entail (b).

a. Each student either laughed or cried.

b. Either each student laughed or each student cried.

Exercise 9.4. Exhibit the two scope readings in FOL translation of *No student likes every teacher*. (In practice speakers do not normally use this S intending the OWS reading. See Szabolcsi (1997) for discussion of the availability of various scope readings).

We now consider some cases of binding and non-binding. Such examples can be tricky. Compare first the two readings of (15). We assume here that *'s mother* is a one place function mapping each individual x to x's mother.

(15) Every child loves his mother.

a. $\forall x(\text{CHILD}(x) \rightarrow \text{LOVE}(x, x\text{'S MOTHER}))$
b. $\forall x(\text{CHILD}(x) \rightarrow \text{LOVE}(x, y\text{'S MOTHER}))$

An occurrence of a variable x in a formula φ is said to be *bound* if it occurs in a constituent of φ of the form $Qx\psi$, where $Q = \forall$ or $\exists$ (or any other VBO). Otherwise that occurrence of x is *free* in φ. All occurrences of x in (15a) and (15b) are bound. But the occurrence of y in (15b) is free. Also we note a technical usage here. In logical parlance *sentences* are the special case of formulas with no free occurrences of variables.

A similar binding vs free pattern is seen in (16a) and (16b) below. Typically argument occurrences of reflexive pronouns (*himself*, *herself*, etc.) in English correspond to bound occurrences of variables in their FOL translations.

(16)

a. Some student criticized himself.
$\exists u(\text{STUDENT}(u) \,\&\, \text{CRITICIZE}(u, u))$
b. Some student criticized him.
$\exists u(\text{STUDENT}(u) \,\&\, \text{CRITICIZE}(u, x))$

In contrast pronouns like *him*, *her*, etc. correspond to free or to bound variables depending on the syntactic context in which they occur. They must be free when their antecedents would be "too close", as in (16b). But when their antecedents are farther away they can be free or bound. So (17a) and (17b) are acceptable FOL translations of (17).

(17) Every teacher likes every student who likes him

a. $\forall x(\text{TEACHER}(x) \rightarrow \forall y((\text{STUDENT}(y) \,\&\, \text{LIKE}(y, x)) \rightarrow \text{LIKE}(x, y)))$
b. $\forall x(\text{TEACHER}(x) \rightarrow \forall y((\text{STUDENT}(y) \,\&\, \text{LIKE}(y, z)) \rightarrow \text{LIKE}(x, y)))$

We note that FOL does not provide a means of directly treating individual constants as VBO's. So its simplest translation of Ss like (18a) is as in (18b), though (18c) has the same truth conditions.

(18) a. John criticized himself.
b. $\text{CRITICIZE}(j, j)$
c. $\exists x(\text{CRITICIZE}(x, x) \,\&\, j = x)$

(18b) is awkward since English does not really like repeated proper nouns as arguments of P_2s. *John criticized John* is felt to be awkward. Speakers tend to say that the two occurrences of *John* refer to different individuals (exactly what we don't represent in (18b)!). Similarly (19a) is not quite adequately translated as (19b), since a free occurrence of x can denote any object in the domain. It might "accidentally" denote John. But English speakers tend to feel that in (19a) the two arguments of *admire* must be different in reference, not simply not necessarily the same.

(19) a. John admires him.
b. $\text{ADMIRE}(j, x)$

Interestingly however repeated occurrences of quantified DPs is not judged awkward: *some student criticized some student*, *just one student praised just one student*, etc. are natural. Observe of course a stark meaning difference, as indicated in the different FOL translations:

(20) a. Every student criticized every student.
a'. $\forall x(\text{STUDENT}(x) \rightarrow (\forall y(\text{STUDENT}(y) \rightarrow \text{CRITICIZE}(x, y)))$
b. Every student criticized himself.
b'. $\forall x(\text{STUDENT}(x) \rightarrow \text{CRITICIZE}(x, x))$

Exercise 9.5. Exhibit an informal model in which (20b') above is true and (20a') false, showing that on the translations given, (20b) does not entail (20a). Note that (20a') does entail (20b').

Exercise 9.6. For each S below provide at least one FOL translation. In some, perhaps many, your translations will be syntactically quite different than the original English S.

a. No one likes everyone who likes himself.
b. John criticized some student other than himself.
c. Every student criticized every student but himself.
d. John criticized every student who did not criticize him.
e. Every teacher either praised or criticized John.
f. At least two students criticized each other.
g. John is a teacher who admires himself.
h. The only person who John criticized was Mary.
i. Every student admires only himself.
j. Only Lucifer admires only himself.

Exercise 9.7. Provide a FOL translation for the S below and argue that it is logically false (= false in every model).

> There is a barber who shaves just those barbers who do not shave themselves.

First Order vs Second or Higher Order Logics

The distinguishing feature of first order logic (and languages) is that we can only quantify over the domain of a model. Second order logic quantifies over subsets of the domain and more generally over functions and relations on the domain, third order logic quantifies over sets of subsets of the domain, etc. Recall for example the sentences we used to define partial orders:

(21) For all binary relations R (over a given domain E)
  a. R is reflexive iff $\forall x(xRx)$
  b. R is antisymmetric iff $\forall x \forall y(xRy \rightarrow yRx)$
  c. R is transitive iff $\forall x \forall y \forall z((xRy \,\&\, yRz) \rightarrow xRz)$

21 is a second order sentence since it quantifies over relations over the domain. A *monadic* second order sentence is one that just quantifies over subsets of the domain.

9.1.4 Interpreting First Order Expressions

We indicated informally how the three types of first order formulas are interpreted. Now we are going to be more explicit about that, as there is one crucial interpretative mechanism we have quietly glossed over. The core idea is that the interpretation of an expression is relativized to "contexts", which we can think of as functions which interpret the free variables in the expression. That is, just as the interpretation of an English sentence such as *She is clever* depends on what the context tells us that *she* refers to, similarly the interpretation of a formula like $3 \leq x$ depends on what the context tells us that x denotes. "contexts" are technically called *assignments (of values to the variables)*. For the moment we are only using individual variables, ones that range over the domain of a model, so an assignment is simply a function from the set VAR $= \{x_1, x_2, \ldots\}$ of variables into E, the domain of the model. We assume an arbitrary first order language $\mathcal{L}$ is given.

Definition 9.1. For all sets E, A_E is [VAR $\rightarrow E$], the set of functions from the variables into E.

And for $\mathcal{M}$ a model with domain E, we design the interpreting function $[\![\cdot]\!]_{\mathcal{M}}$ so that expressions denote functions from the set A_E of assignments into the appropriate denotation set. An expression with no free variables will denote a constant function, that is, its interpretation does not vary with the context (assignment). For example a lexical P_1 like *laugh* has no free variables and so will denote a constant function from A_E into $[E \to \{T, F\}]$. On our old way of interpreting *laugh* it simply denoted an element of $[E \to \{T, F\}]$, so making it a function which associates a fixed element of $[E \to \{T, F\}]$ with all the contexts is not really very different. The important difference shows up with expressions that have free occurrences of variables.

A formula like x *laughed* will denote, not a truth value, but a function from contexts (assignments) to truth values. And its truth value may vary with the assignment (context). For example if α is an assignment that maps x to John and β is one that maps x to Jane then the denotation of x *laughed* might map α to T (if John laughed) and β to F (if Jane didn't laugh). In this way the interpretation of "open" expressions—ones with free variables—may vary with the context according to what the free variables denote.

And the crucial place where these assignments come into play is with the interpretation of quantified formulas. To give the main idea we first define:

Definition 9.2. For all assignments α and β, all variables x, β is an *x-variant* of α iff $\beta(y) = \alpha(y)$ for all variables $y \neq x$. So β may assign a different value to x than α does (though it doesn't have to) but it assigns the same value as α does to all the other variables. For b in the domain of the model $\mathcal{M}$, we write $\alpha^{x \to b}$ for that x-variant of α which maps x to b. Formally

$$\alpha^{x \to b}(y) =_{df} \begin{cases} \alpha(y) & \text{if } y \neq x \\ b & \text{if } y = x. \end{cases}$$

Note that for each variable x, the binary relation *is an x-variant of* is an equivalence relation on A_E. (Reflexivity and symmetry are trivial. For transitivity let x be an arbitrary variable and suppose that α is an x-variant of β and β an x-variant of γ. We must show that α is an x-variant of γ. Let y be a variable other than x. Then $\alpha(y) = \beta(y)$ by the first assumption, and $\beta(y) = \gamma(y)$ by the second. So by the transitivity of $=$, $\alpha(y) = \gamma(y)$ showing that α is an x-variant of γ). Now we give the truth conditions for quantified formulas as follows:

Definition 9.3. For all models $\mathcal{M}$ with domain $E_\mathcal{M}$, and all assignments α,

a. $[\![\forall x\varphi]\!]_\mathcal{M}(\alpha) = T$ iff for all $b \in E_\mathcal{M}$, $[\![\varphi]\!]_\mathcal{M}(\alpha^{x \to b}) = T$, and
b. $[\![\exists x\varphi]\!]_\mathcal{M}(\alpha) = T$ iff for some $b \in E_\mathcal{M}$, $[\![\varphi]\!]_\mathcal{M}(\alpha^{x \to b}) = T$.

Thus φ universally quantified maps a context α to T iff φ maps to T every context that differs from α at most by what it assigns to the variable x. Similarly φ existentially quantified is true in a context α iff φ is true of some assignment that differs from α at most in what it assigns to x. In both cases the interpretation of the quantified formula is done compositionally: $[\![Qx\varphi]\!]_{\mathcal{M}}$ is defined in terms of $[\![\varphi]\!]_{\mathcal{M}}$, all $Q = \forall$ or $\exists$.

For later reference we give a comprehensive formal definition of model and entailment for first order languages.

Definition 9.4. For $\mathcal{L}$ an arbitrary first order language a model $\mathcal{M}$ is a pair $(E, [\![\cdot]\!]_{\mathcal{M}})$, where E is a non-empty set, the domain of $\mathcal{M}$, and $[\![\cdot]\!]_{\mathcal{M}}$ is a function mapping each expression of $\mathcal{L}$ to a function from the assignments over E satisfying:

a. Lexical conditions:
   i. For P an n-place predicate symbol, $[\![P]\!]_{\mathcal{M}}$ is a constant function from A_E into $[E^n \to \{T, F\}]$.
   ii. For c an individual constant $[\![c]\!]_{\mathcal{M}}$ is a constant function from A_E into E.
   iii. For F an $n > 0$ place function symbol, $[\![F]\!]_{\mathcal{M}}$ is a constant function from A_E into $[E^n \to E]$.

b. Term conditions:
   i. For x a variable $[\![x]\!]_{\mathcal{M}}$ maps each assignment α to $\alpha(x)$ (So in a context α a variable denotes what the context says it denotes)
   ii. For F an n-place function symbol and $t_1, \ldots, t_n$ terms, $[\![F(t_1, \ldots, t_n)]\!]_{\mathcal{M}}$ maps each assignment α to $[\![F]\!]_{\mathcal{M}}(\alpha)([\![t_1]\!]_{\mathcal{M}}(\alpha), \ldots, [\![t_n]\!]_{\mathcal{M}}(\alpha))$.
   (That is, $[\![F(t_1, \ldots, t_n)]\!]_{\mathcal{M}}$ maps each assignment α to the object that the function F is interpreted as in α maps the n-tuple of objects that the n terms are interpreted as in α.)

c. Atomic formula conditions:
   i. For P an n-place predicate symbol and $t_1, \ldots, t_n$ terms, $[\![P(t_1, \ldots, t_n)]\!]_{\mathcal{M}}$ maps each assignment α to $[\![P]\!]_{\mathcal{M}}(\alpha)([\![t_1]\!]_{\mathcal{M}}(\alpha), \ldots, [\![t_n]\!]_{\mathcal{M}}(\alpha))$.

d. Boolean conditions:
   Boolean compounds are interpreted pointwise on the assignments
   i. $[\![\varphi \,\&\, \psi]\!]_{\mathcal{M}}(\alpha) = [\![\varphi]\!]_{\mathcal{M}}(\alpha) \wedge [\![\psi]\!]_{\mathcal{M}}(\alpha)$.
   ii. $[\![\varphi \text{ or } \psi]\!]_{\mathcal{M}}(\alpha) = [\![\varphi]\!]_{\mathcal{M}}(\alpha) \vee [\![\psi]\!]_{\mathcal{M}}(\alpha)$.
   iii. $[\![-\varphi]\!]_{\mathcal{M}}(\alpha) = \neg[\![\varphi]\!]_{\mathcal{M}}(\alpha)$.
   iv. $[\![\varphi \to \psi]\!]_{\mathcal{M}}(\alpha) = \neg[\![\varphi]\!]_{\mathcal{M}}(\alpha) \vee [\![\psi]\!]_{\mathcal{M}}(\alpha)$.
   v. $[\![\varphi \leftrightarrow \psi]\!]_{\mathcal{M}}(\alpha) = T$ iff $[\![\varphi]\!]_{\mathcal{M}}(\alpha) = [\![\psi]\!]_{\mathcal{M}}(\alpha)$.

e. Quantifier conditions:

Quantified formulas are interpreted as in Definition 9.3 above. An equivalent statement is:

i. $[\![\forall x\varphi]\!]_{\mathcal{M}}(\alpha) = \bigwedge\{[\![\varphi]\!]_{\mathcal{M}}(\alpha^{x\to b}) \mid b \in E\}$
(Recall, for K a non-empty subset of $\{T, F\}$, $\bigwedge K = T$ iff $K = \{T\}$.)

ii. $[\![\exists x\varphi]\!]_{\mathcal{M}}(\alpha) = \bigvee\{[\![\varphi]\!]_{\mathcal{M}}(\alpha^{x\to b}) \mid b \in E\}$
(Recall, for $K \subseteq \{T, F\}$, $\bigvee K = T$ iff $T \in K$.)

So the universal quantifier, like *and*, is a glb operator, and the existential quantifier, like *or*, is a lub operator. Given a context, $\forall x\varphi$ denotes the greatest lower bound of the set of truth values denoted by φ when we let x denote successively the elements of E. (If all those values are T then the set is $\{T\}$ and its glb is T. But if one of those values is F then the set is $\{F\}$ or $\{F, T\}$ and in each case its glb is F). In a similar way existential quantification is a least upper bound operator. $\exists x\varphi$ denotes the least upper bound of the set of truth values denoted by φ when we let x denote successively the elements of E. If one those values is T then the set is $\{T\}$ or $\{T, F\}$ and in each case its lub is T. If none of its values is T then the set is $\{F\}$ and its lub is F.

Notational variants. Texts often write $[\![\varphi]\!]^{\alpha}_{\mathcal{M}}$ or $[\![\varphi]\!]^{\mathcal{M},\alpha}$ instead of $[\![\varphi]\!]_{\mathcal{M}}(\alpha)$ and read it as "the interpretation of φ in the model $\mathcal{M}$ at the assignment α". We will stay with our function-argument notation but the reader should realize that anything that can be said with one notation can be said with the other.

Now to define truth in a model and entailment it simplifies matters to use the Coincidence Lemma below, which guarantees that we can ignore assignments when dealing with expressions that have no free variables. We first define what we mean by a *free variable* in φ: FV will be a function mapping each expression φ to the set of variables which occur free in φ:

Definition 9.5.

a. $\mathrm{FV}(c) = \emptyset$ if c is an individual constant,
b. $\mathrm{FV}(x) = \{x\}$ if x is a variable,
c. $\mathrm{FV}(H(t_1, \ldots, t_n)) = \mathrm{FV}(t_1) \cup \cdots \cup \mathrm{FV}(t_n)$ if H is an n-place function or predicate symbol and $t_1, \ldots, t_n$ are terms,
d. $\mathrm{FV}(-\varphi) = FV(\varphi)$,
e. $\mathrm{FV}(\varphi\, c\, \psi) = \mathrm{FV}(\varphi) \cup \mathrm{FV}(\psi)$ for $c = \&, \mathrm{or}, \to$, or $\leftrightarrow$, and
f. $\mathrm{FV}(\forall x\varphi) = \mathrm{FV}(\exists x\varphi) = \mathrm{FV}(\varphi) - \{x\}$ for x any variable.

Theorem 9.1 (The Coincidence Lemma)**.** *For φ a first order formula, $\mathcal{M}$ a model and α and β assignments, if $\alpha(x) = \beta(x)$ for all variables*

$x \in FV(\varphi)$, then $[\![\varphi]\!]_{\mathcal{M}}(\alpha) = [\![\varphi]\!]_{\mathcal{M}}(\beta)$.

Corollary 9.2. *If φ has no free variables ($FV(\varphi) = \emptyset$) then for all assignments α, β, $[\![\varphi]\!]_{\mathcal{M}}(\alpha) = [\![\varphi]\!]_{\mathcal{M}}(\beta)$.*

Definition 9.6. For φ a formula, $\mathcal{M} = (E, [\![\cdot]\!]_{\mathcal{M}})$ a model and α an assignment,

a. φ is *True in* $\mathcal{M}$ *at* α iff $[\![\varphi]\!]_{\mathcal{M}}(\alpha) = T$. In such a case we also write $\mathcal{M} \models_{\alpha} \varphi$ and say that $\mathcal{M}$ *satisfies* φ at α.

b. If φ is a sentence (no free variables recall), φ *is True in* $\mathcal{M}$, noted simply $[\![\varphi]\!]_{\mathcal{M}} = T$, iff for some α, $[\![\varphi]\!]_{\mathcal{M}}(\alpha) = T$. In such a case we write simply $\mathcal{M} \models \varphi$ and say that $\mathcal{M}$ *satisfies* φ. (If a sentence φ is True at some α it is True at all, by the Coincidence Lemma). φ is said to be *satisfiable* iff there a model $\mathcal{M}$ such that $\mathcal{M} \models \varphi$ and φ is *logically true* iff for all models $\mathcal{M}$, $\mathcal{M} \models \varphi$.

c. A set K of sentences *entails* ($\models$) a sentence ψ iff for all $\mathcal{M}$, $\bigwedge\{[\![\varphi]\!]_{\mathcal{M}} \mid \varphi \in K\} \leq [\![\psi]\!]_{\mathcal{M}}$. (This just says that ψ is true in all models in which the φ in K are simultaneously True). We say that a sentence φ *entails* a sentence ψ iff $\{\varphi\} \models \psi$. And sentences φ and ψ are *logically equivalent*, noted $\varphi \equiv \psi$, iff each entails the other.

9.1.5 Remarks

Notational simplification. Generalizing our convention in Definition 9.6b above, we write simply $[\![\varphi]\!]_{\mathcal{M}}$ instead of $[\![\varphi]\!]_{\mathcal{M}}(\alpha)$ when φ of any category is *closed*, that is, has no free variables and thus denotes a constant function on the assignments. And when φ is syntactically simple we often omit the double brackets, writing simply $\varphi_{\mathcal{M}}$ instead of $[\![\varphi]\!]_{\mathcal{M}}$. For example if EVEN is a one place predicate symbol in a language (say, the language of elementary arithmetic) we would write $\text{EVEN}_{\mathcal{M}}$ instead of $[\![\text{EVEN}]\!]_{\mathcal{M}}$.

Some simple logical equivalences. Here are some useful logical equivalences involving the quantifiers and negation:

(22)
a. $\forall x\varphi \equiv -\exists x - \varphi$
b. $\exists x\varphi \equiv -\forall x - \varphi$
c. $-\forall x\varphi \equiv \exists x - \varphi$
d. $-\exists x\varphi \equiv \forall x - \varphi$

The universal and existential quantifiers are *duals* of each other, just as *and* and *or* are (and just as $\Box$ (necessity) and $\Diamond$ (possibility) are in modal logic). The interpretations of *and*, *all*, $\Box$ are meet operations, and those for *or*, *some*, and $\Diamond$ are join operations.

The rules which prefix a quantifier to a formula φ do not require that the variable they use occur in φ (much less occur free). In linguistic parlance first order languages allow *vacuous quantification*. It is sometimes held that natural languages disallow it. In any event, (23a,b) are well formed formulas in the language of first order arithmetic:

(23) a. $\forall x(0 < 1)$
b. $\exists x(0 > 1)$

Moreover such formulas pose no interpretative problem, not even a special case. If x has no free occurrences in φ then

$$[\![\forall x\varphi]\!]_{\mathcal{M}} = [\![\exists x\varphi]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}.$$

This follows from our truth definition for quantified formulas (and uses the Coincidence Lemma). Still Ss such as (23a,b) seem useless, so why don't we restrict the syntax of FOL to require that Qx can only combine with a φ in which x occurs free? The answer is that such a restriction would force us to make many unenlightening restrictions on meta-theorems. For example it is a meta-theorem that universal quantification distributes over conjunction:

$$\forall x(\varphi \,\&\, \psi) \equiv (\forall x\varphi \,\&\, \forall x\psi).$$

Now if x was not free in φ but was in ψ it would occur free in $(\varphi \,\&\, \psi)$. So the formula on the left of $\equiv$ is well formed, but the one on the right is not, since $\forall x\varphi$ is not a formula. So the natural distributivity meta-theorem doesn't hold in the more restrictive syntax, and the reason does not seem enlightening.

Alphabetic variants. Every first order formula has infinitely many *alphabetic variants*, all logically equivalent to it. For example (24a,b,c) differ just by choice of bound variable, they are alphabetic variants and logically equivalent. Definition 9.7 is the explicit definition.

(24) a. $\forall x(Px \to Qx)$
b. $\forall y(Py \to Qy)$
c. $\forall z(Pz \to Qz)$

Definition 9.7. For $\mathcal{L}$ a first order language with V its basic vocabulary and VAR its set of individual variables, let π be a permutation of VAR (so π is a bijection from VAR onto VAR). Extend π to V by putting $\pi(d) = d$, all $d \in V$. Extend π to all sequences of symbols $s = \langle s_1, \ldots, s_n\rangle$ over $V \cup$Var by setting $\pi(s) = \langle \pi(s_1), \ldots, \pi(s_n)\rangle$. Then we define a formula ψ to be an *alphabetic variant* of a formula φ iff for some such π, $\psi = \pi(\varphi)$.

Exercise 9.8. Show that the *is an alphabetic variant of* relation on expressions is an equivalence relation.

One shows by induction on formula and term complexity that any two alphabetic variants are logically equivalent. Here is the induction principle on formulas that would be used:

Definition 9.8. Let $\mathcal{L}$ be the set of expressions of some first order language. Then if $K \subseteq \mathcal{L}^*$ satisfying conditions (a)–(c) below then all formulas of $\mathcal{L}$ are in K:

a. All atomic formulas are in K.
b. K is closed under the formation of boolean compounds. That is, if $\varphi, \psi \in K$ then $-\varphi$, $(\varphi \,\&\, \psi), (\varphi \text{ or } \psi), (\varphi \rightarrow \psi)$ and $(\varphi \leftrightarrow \psi)$ are all in K.
c. K is closed under universal and existential quantification. That is, if $\varphi \in K$ then for all variables x, $\forall x\varphi$ and $\exists x\varphi \in K$.

Thus to prove that all formulas have some property P let K be the set of formulas with P and show that all atomic formulas are in K and that K is closed under the formation of boolean compounds and quantification.

We turn in a moment to one extension to the class of VBO's—the *lambda* operator—which makes the logic a more natural vehicle for representing natural language. But let us note some general linguistic properties of FOL. Little that we do later in this book presupposes this material, but FOL is well studied and has many appealing linguistic properties, to the point where it is not foolish to think of mathematical logic as a mode of linguistic analysis—but the languages studied are mathematical ones (the language of Set Theory, Euclidean Geometry, Elementary Arithmetic for example) and increasingly programming languages in computer science.

9.2 Some General Linguistic Properties of First Order Logic (FOL)

Three of the four general properties of SL (Sentential Logic) we presented in Chapter 7 generalize to FOL, though the generalizations involve significant enrichments in some cases.

Proposition 9.3. *Both SL and any first order language are* compact*: whenever a set K of sentences entails a sentence φ then some finite subset of K entails φ.*

Both SL and all first order languages satisfy *interpolation*, though its statement must be enriched to take into account the richer syntax of FOL (which includes the logical constant $=$) and its proof is significantly more complex (see Boolos and Jeffrey (1980) Ch. 23 and Monk (1976) Ch. 22):

Proposition 9.4 (Craig's Interpolation Lemma)**.** *For $\mathcal{L}$ a first order language and φ, ψ sentences of $\mathcal{L}$, if $\varphi \models \psi$ then there is a sentence τ of $\mathcal{L}$ such that*

a. $\varphi \models \tau$ *and* $\tau \models \psi$ *and*

b. all non-logical constants occurring in τ occur in both φ and ψ.

As in SL the Craig Lemma is a kind of relevancy condition on entailment. It says that whether some ψ is (non-trivially) entailed by some φ depends just on what they have in common.

Soundness and Completeness generalize to the FOL case. Since we have a greater diversity of syntactic structures in FOL (predicates and arguments, quantifiers) we must extend the (syntactically defined) deduction system with more derivational rules. For example, if we have a line in a proof of the form $\forall x(Px)$ and t is a term in our language, then we must be able to infer $P(t)$. But logicians have formulated (in several ways) the deduction system(s) we need, yielding the following elegant theorem, where $S \vdash \varphi$ says that there is a proof of φ from premises in S:

(25) $S \models \varphi$ iff $S \vdash \varphi$

So again the semantic relation, entailment, is syntactically characterized by $\vdash$ relation. The left to right direction of (25) is *completeness*: if φ follows from S then there is a proof of φ from premises in S. The right to left direction is *soundness*—whenever we can derive φ from a set of premises S then indeed S really does entail φ. Also (25) enables us to give a linguistically useful equivalent statement of compactness. Namely,

(26) For any set K of sentences in a first order language $\mathcal{L}$, K has a model iff every finite subset of K has a model.

(A set of sentences has a model $\mathcal{M}$ iff every $\varphi \in K$ is true in $\mathcal{M}$ (that is $[\![\varphi]\!]_{\mathcal{M}} = T$)). To prove (26) from compactness as we defined it, note that the left to right direction above is trivial: if some $\mathcal{M}$ makes every sentence in K true then for any finite subset K' of K, every sentence in K' is true in $\mathcal{M}$. Going the other way, suppose that every finite subset of K has a model. And leading to a contradiction assume that K itself does not have a model. Then (vacuously) $K \models \exists x(x \neq x)$, the latter sentence having no models, so by completeness $K \vdash \exists x(x \neq x)$, and since proofs are finite sequences of formulas only finitely many of the sentences in K were used in the proof. Let K' be such a set. So $K' \vdash \exists x(x \neq x)$ whence by soundness $K' \models \exists x(x \neq x)$. But that means that K' has no model since any model of it would also have to be a model of $\exists x(x \neq x)$. And this contradicts our assumption that every finite subset of K has a model. Thus compactness entails the displayed sentence above (and in fact is equivalent to it, but we only need the entailment case later).

Definability (Is English first order?). Here we are concerned with whether the semantic analysis of English (or any other natural language) can be given in the first order apparatus we spelled out above. Coming at this question cold a "Yes" answer would seem discouraging. Thorough grammars of natural languages run to several hundreds of pages, and they don't even pretend to enumerate just the expressions competent speakers accept, nor, despite semantic insights, do they provide a systematic semantic interpretation of the sort we gave for FOL above in just a few pages. Is it really plausible that we only need a page and a half of semantic interpretation to represent what it takes us hundreds of pages to spell out in the syntax? Nonetheless at a certain point in the history of generative grammar linguists tended to assume that something close to a first order semantics for a natural language would suffice (See Chomsky and Lasnik (1977) for such an assumption, albeit one that played no important role in their conclusions in that article). Then several scholars debated whether certain (sometimes subtle) aspects of English were or were not first order definable (Barwise (1979), Hintika (1974), Gabbay and Moravcsik (1974), Guenthner and Hoepelmann (1974), Fauconnier (1975), Boolos (1981)). Below we offer a non-subtle argument that English is not first order: we exhibit a class of naturally expressible quantifiers which provably cannot be defined in FOL.

But first, idle curiosity aside, "Why should we care whether English semantics can be expressed in first order?". One reason is that FOL has many very nice properties, of which we have begun to cite a few. So if the syntax and semantic interpretation of English (or any other natural language) can be given in first order that means that English and presumably natural languages in general have these nice properties. Perhaps more important scientifically, FOL is well studied and well understood. So if we succeed in formulating English semantics in first order we will have succeeded in representing something we are trying to understand in terms of something we already understand. To some extent,

Knowledge is translation from the unknown to the known.

Thirdly, as we have noted above, many mathematical theories have or admit of first order axiomatizations (meaning that the axioms of the theory are first order sentences). Set theory is a crucial case in point, as it is often held that most of mathematics can be coded in set theory. Now regardless of the precise status of this latter claim, it is clear that we can express a lot in first order formulas, so it would be interest to learn whether the semantic analysis of natural languages forces us beyond that or not. To support such claims, we must first be clear about what it means to be first order definable. Let us give some examples, both pro and con.

(27) a. *There exist at least n things* can be said in first order (for each $n \in \mathbb{N}$), but
b. *There exist just finitely many things* is not sayable in first order, nor is *There are infinitely many things.*

Here is how we say *There are at least 3 things.* Let φ be defined by

$$\varphi = \exists x \exists y \exists z (x \neq y \,\&\, x \neq z \,\&\, y \neq z).$$

φ is clearly a first order sentence and for all models $\mathcal{M}$, $[\![\varphi]\!]_{\mathcal{M}} = T$ iff $|E_{\mathcal{M}}| \geq 3$. A common way of putting this would be,

(28) for all models $\mathcal{M}$, $\mathcal{M} \models \varphi$ iff $3 \leq |E_{\mathcal{M}}|$.

So the models that satisfy φ are just those whose domains have at least three elements. And in general for any $n \in \mathbb{N}$, we can say *There are at least n things* in a comparable way: we begin with n existential quantifiers using distinct variables and form the conjunction of all the non-equals sentences using all distinct pairs of these variables. So all we need is the logical predicate =.

Now consider (27b). Negative claims like (27b) are harder to establish than positive ones like (27a) since we have to show that no first order formula φ is true in a model iff its domain is finite. Intuitively the sentence we need would have the meaning of an infinite disjunction: *There is at least one thing or there are at least two things or ... at least three things*, etc. But first order languages do not have infinitely long sentences, so this is not a possibility. To prove that there is no first order sentence with the property we want we use the compactness property of FOL noted above (and which may have seemed unduly "mathematical" on first pass).

Proof. Let $\Phi = \{\varphi_n \mid n \in \mathbb{N}\}$, where φ_n is the sentence *There exist at least n things.* Let ψ be the sentence *There exist just finitely many things.* Then any finite subset K of $\Phi \cup \{\psi\}$ has a model. Just choose $E_{\mathcal{M}}$ to be $\{1, 2, \ldots, n\}$ where n is the largest number such that $\varphi_n \in K$.

The model is finite so ψ is true (whether it is in K or not) and each φ_m in K is true in $\mathcal{M}$. Thus every finite subset of $\Phi \cup \{\psi\}$ has a model, so by compactness (if our language is first order) the entire set $\Phi \cup \{\psi\}$ has a model $\mathcal{M}$. But it doesn't since if ψ holds $\mathcal{M}$ is finite, of cardinality m say, so φ_{m+1} is false in $\mathcal{M}$ so $\mathcal{M}$ is not a model for $\Phi \cup \{\psi\}$. So a language which can say *There are just finitely many things* is not first order. $\square$

As an easy corollary we infer that *There are infinitely many things* is also not a first order statement, otherwise its negation, *There aren't infinitely many things*, would be a first order statement that meant *There are just finitely many things*. And from these observations it is a small step to show that the quantifiers in (29) are not first order definable:

(29) a. *Just finitely many* sentences have fewer than ten words.
b. *Infinitely many* sentences have more than ten words.
c. *All but finitely many* sentences have more than ten words.

Let us say explicitly what it means for an English-type quantifier (including the mathematical ones above) to be first order definable. In the simplest cases in English quantifiers such as *all*, *some*, *no*, *most*, *several*, *not all*, *most but not all*, *nearly forty and infinitely many others* (Chapter 11) are Determiners which combine with a common noun such as *poet* to form a DP, which combines in the basic case with a P_1 to form a P_0, as in *All (most, no, some) poets daydream*.

Definition 9.9. For D a functional which associates with each domain E a function D_E from $\mathcal{P}(E)$ into $[\mathcal{P}(E) \to \{T, F\}]$, D is *first order* iff there is a sentence φ in a first order language whose only non-logical constants are two one place predicate symbols P,Q such that for all models $\mathcal{M}$ for $\mathcal{L}$,

$$\mathcal{M} \models \varphi \text{ iff } D_{E_{\mathcal{M}}} P_{\mathcal{M}} Q_{\mathcal{M}} = T.^2$$

For example to see that *exactly one* is first order, definable set

$$\varphi = \exists x((Px \,\&\, Qx) \,\&\, \forall y((Py \,\&\, Qy) \to y = x)).$$

And taking EXACTLY ONE to map sets A, B to T iff $|A \cap B| = 1$ we can prove that $(\text{EXACTLY ONE})_E P_M Q_M = T$ iff $[\![\varphi]\!]_{\mathcal{M}} = T$, showing that φ defines this quantifier.

If the only English quantifiers that lay beyond the first order boundary used technical notions like *(in)finite* we might not, as linguists,

[2] $P_{\mathcal{M}}$ here is the set of $b \in E$ that $[\![P]\!]_{\mathcal{M}}$ maps to T (under any assignment). Analogously for $Q_{\mathcal{M}}$.

regard the boundary as very constraining. So we note a few additional cases, the first showing that first order definability may be subtle in appearance (unlike the unsubtle appearance of words like *finite*, etc.).

First, quantifying over natural numbers is usually not first order definable and some statements that do that do not do it in an obvious way. Adapting informally an example from Kolaitis (2006) to whom we refer the reader for a precise statement and proof, let us consider the set $T[r, b]$ of trees with r the root and b a leaf node distinct from r. Let φ_n be the statement *There is no path of length* n *or less from* r *to* b. Set $\Phi = \{\varphi_n \mid n \in \mathbb{N}\}$ and let ψ say For all distinct nodes x, y there is a path from x to y or a path from y to x. (Recall that a path was defined to be a finite sequence of distinct nodes satisfying certain (first order) conditions). Let K be any finite subset of $\Phi \cup \{\psi\}$. Clearly K has a model: just choose the tree to be a chain with n distinct nodes between r and b. The path from r to b has length $n + 1$ and that is the only path from r to b so there is no path of length n from r to b. Further ψ obviously holds in every chain. By compactness $\Phi \cup \{\psi\}$ should have a finite model. But it doesn't, as $\Phi \cup \{\psi\} \models \varphi_n$ for every $n > 0$. (Kolaitis shows a stronger result—connectivity is not first order definable even over finite models).

Our last examples are again drawn from natural language proper. The first concerns proportionality quantifiers like *most* in the sense of *more than half.* Such quantifiers include fractional and percentage expressions such as *a third of*, *seventy per cent of*, as well as a variety of constructions whose meaning is proportional but are not constructed with *of* as partitives are: *seven out of ten (sailors smoke Players)*, *not one student in ten (knows the answer to that question)*. Now Barwise and Cooper (1981) provide a summary argument that *more than half* is not first order definable (even limiting ourselves to finite domains). The techniques introduced in Westerståhl (1989)) can also be used. And in general when D is properly proportional (D is not a boolean function of 100% (= ALL) and 0% (= NO), and the truth of D$(A)(B)$ depends on the proportion of As that are Bs) D is not first order definable. Without reconstructing the "back and forth" methods used in the proof, we may help convince the reader of the non-first-orderizability of *most* and its kin by observing that we could explicitly enumerate the cases of *Most* A*s are* B*s* if there were just finitely many As and we knew how many. For example, for $|A| = 5$, *Most* A*s are* B*s* is true iff *at least three of the five* A*s are* B*s*. And *at least 3 of the 5*, and more generally *at least* n *of the* m, for $n, m \in \mathbb{N}$, are first order definable. So intuitively most could be expressed by a disjunction of the form: *at least two of the three or at least three of the four or at least three of the five,...* but, as with *finite*, we cannot form an infinite disjunction in first order. So we see that proportionality quantifiers pose problems comparable to those of more mathematical predicates.

Moreover the proportionality cases also serve to show that cardinal comparisons are not definable in first order. Such comparatives are illustrated in (30). Note that they combine with two Nouns and a P_1 and so denote functions mapping three subsets of the domain into $\{T, F\}$.

(30) a. *More* students *than* teachers came to the party.
b. John interviewed *exactly as many* men as women.
c. *More than twice as many* men *as* women get drafted.

If *more...than...* were first order definable then we could define *most* by saying *Most As are Bs* iff *More As that are Bs than As that are not Bs exist*. But since *most* in this sense is not first order definable neither is *more...than...* or the infinitely many other cardinal comparatives.

Decidability however does not fully carry over from SL to FOL. There is no algorithm that tells us for an arbitrary first order sentence φ that it is true in all models if it is, and that it isn't, if it isn't (Church (1936), Turing (1936)). On reflection this is not surprising. Many quite non-trivial mathematical theories, such as Set Theory, Group Theory, and Elementary Arithmetic, have or can be given axioms in first order. And many proofs from these axioms seem non-obvious, ingenious, or complicated. So it would be surprising to learn that we could program a computer to look at any first order formula we wrote down and tell us that it is true if it is and that it is false if it isn't.

On the other hand, while validity is not decidable in FOL, it does have a weaker property—that of being *semi-decidable* (Enderton (1985)), also called in this case *recursively enumerable*. By the completeness of FOL we know that if a first order φ is valid then there is a proof of φ (from no premises). A proof is a finite sequence of formulas and whether a finite sequence of formulas is a proof or not is decidable. Thus there is a mechanical procedure which tells us that an arbitrary φ is valid if it is, but no procedure which tells us that φ is not valid if it isn't. We note that if a set, say the set of valid formulas in some first order language, is recursively enumerable and its set theoretic complement is also then the set itself is decidable (also called *recursive*).

In fact, not only is validity (truth in all models) undecidable in FOL, so is satisfiability: there is no algorithm that will say of an arbitrary first order sentence whether it has a model or not.

There are however some special cases, of some linguistic interest, in which decidability is restored. Here are two:

(31) a. FOL with at most two variables is decidable (Mortimer (1975), Grädel et al. (1997)). Decidability is lost when we allow even three variables (Kahr et al. (1962)).
b. Monadic First Order Logic is decidable (Boolos and Jeffrey (1980) Ch. 25) but FOL with even one $n > 1$ place predicate is not decidable (Boolos and Jeffrey (1980) Ch. 22).

Concerning (31a), some axioms for familiar theories crucially use formulas in three variables. The statement of *distributivity* in the defining conditions for boolean lattices is one such. Recall, that a lattice $(L, \leq)$ is distributive iff

(32) a. $\forall x, y, z(x \wedge (y \vee z)) = ((x \wedge y) \vee (x \wedge z))$, and
b. $\forall x, y, z(x \vee (y \wedge z)) = ((x \vee y) \wedge (x \vee z))$.

Similarly the claim that an order relation is transitive, (21c) or that a binary function (like $\cap$, $\wedge$) is associative $((A \cap B) \cap C) = (A \cap (B \cap C))$ uses three variables in an essential way. And impressionistically when verifying that a given mathematical structure satisfies various conditions–say that some given partial order relation meets the conditions for being a boolean lattice—it is the conditions using three variables which are the hardest to verify. (31a) is a logical correlate of this impression.

Concerning (31b), a language $\mathcal{L}$ is *monadic* first order if all its predicates are at most unary (so it has no n-place predicate symbols for $n > 1$). For such an $\mathcal{L}$ there is an algorithm which tells us for any sentence φ in $\mathcal{L}$ whether it is valid or not. Decidability here hinges on the fact that Monadic FOL has the *finite model property*: If φ is false in some model then it is false in a model with a finite domain, where an upper bound on the size of the domain can be computed as a function of the syntactic structure of φ—$2^k \cdot r$ will do, k the number of unary predicates in φ and r the number of variables (Boolos and Jeffrey (1980)). So to verify φ we "merely" check the finite number of models of that size or less. If φ is not false in any of them then it is valid (logically true). Otherwise it is not.

This apparently technical fact is of some linguistic interest. One might have thought that the reason it is harder to evaluate whether a formula is valid in FOL compared to SL is due to the quantifiers, which allow us to make claims about arbitrarily many objects, in particular about infinitely many. And doubtless this is where some of the complexity of FOL comes from. But not all of it. Monadic FOL has the full range of quantifiers and variables as in full FOL and validity is decidable. But adding a single two place predicate symbol to Monadic FOL results in a loss of decidability (Boolos and Jeffrey (1980) Chs. 22, 25).

So we see that having two (and greater) place predicates in our language significantly increases logical complexity. And in the syntactic and semantic analysis of natural language many of the phenomena we study are only, or primarily, significant when transitive (and ditransitive) verbs are considered. For example, the basic concern of Binding Theory is mostly of interest when we are binding co-arguments of a given predicate. If English had only one place predicates reflexive pronouns would probably not exist. **Himself walks* is ungrammatical, it seems, as is **Himself criticized Dana*, but *Dana criticized himself* is fine, as are *Dana sent himself flowers* and *He often treats himself to a fine cognac after dinner*. Equally morphological causatives (Turkish, Malagasy, Tsez) primarily make P_2s from P_1s (and also often P_3s from P_2s):

(33) Malagasy; Austronesian:

a.

mihomehy	*izy*
laugh	3nom

'He is laughing.'

b.

mampihomehy	*azy*	*izy*
cause+laugh	3acc	3nom

'He is making him laugh.'

Similarly Passive primarily derives P_1s from P_2s and P_2s from P_3s. In some languages (German, Turkish, Latin) it may in addition derive P_0s from P_1s. (34) and (35) exhibit P_1 and P_2 derivations by Passive in Kinyarwanda (Bantu).

(34) a.

umugore	[$_{P_2}$ *a-ra-andik-a*]	*ibaruwa*
woman	she-pres-write-asp	letter

'The woman is writing a letter.'

b.

Ibaruwa	[$_{P_1}$ *i-ra-andik-w-a*]	*(n'umugore)*
letter	it-pres-write-Pass-asp	(by'woman)

'The letter is being written (by the woman).'

(35) a.

umugabu	[$_{P_3}$ *y-a-haa-ye*]	*umugore*	*igitabo*
man	he-past-give-asp	woman	book

'The man gave the woman the book.'

b.

umugore	[$_{P_2}$ *y-a-haa-w-e*]	*igitabo*	*(n-umugabo)*
woman	she-past-give-Pass-asp	book	(by man)

'The woman was given the book (by the man).'

c.

igitabo	[$_{P_2}$ *cy-a-haa-w-e*]	*umugore*	*(n'umugabo)*
book	it-past-give-Pass-asp	woman	(by man)

'The book was given to the woman by the man.'

And lastly, case marking and verb agreement paradigms, while serving a variety of functions, are most prominent with P_2s and P_3s, where understanding requires that speakers be able to identify which DPs bind which arguments of the predicate. There is no issue if the predicate is monadic—one DP, one argument to bind. But with P_2s if what we hear in some language is *praise Mary Sue* we need a way to decide if it means *Sue praised Mary* (per the word order conventions in Fijian and Tzotzil (Mayan)) or it means *Mary praised Sue* (per the word order conventions in Berber and Maasai). In languages with extensive case marking, such as Latin, Warlpiri (Australia) and even Japanese, word order may be fairly free preverbally, and it is the case marking conventions we rely on to associate DPs with semantic arguments of the predicates.

The non-utility of limiting ourselves to finite models. We might have expected that we could simplify the task of semantic evaluation and restore decidability by limiting ourselves to finite models (ones whose domains are finite). Surely part of the undecidability results depends on the fact that to evaluate the truth of a quantified sentence we may have to search through an infinite domain. But it turns out that restricting ourselves to finite domains doesn't help:

Theorem 9.5 (Trakhtenbrot (1950))**.** *Neither validity nor satisfiability are decidable in FOL even if we restrict attention just to models with finite domains.*

In fact matters get worse. Trakhtenbrot's theorem tells us that there isn't even a proof procedure for the set of formulas true in all finite models. So the set of first order formulas true in all finite models is not even recursively enumerable. See Lassaigne and de Rougement (1993) pp. 177-180 for an exposition.

We end this section by discussing a characterization of FOL which shows that we can not significantly increase its expressive power without losing some of its "nice" properties, such as compactness or completeness and Löwenheim-Skolem (below).

Theorem 9.6 (Löwenheim-Skolem)**.** *For K a set of first order sentences, if K has an infinite model then K has models in every cardinality greater than or equal to $|K|$.*

Loosely this theorem says that first order formulas can't discriminate among different infinite cardinals. (Recall that $|\mathbb{N}| < |\mathcal{P}(\mathbb{N})| < |\mathcal{P}(\mathcal{P}(\mathbb{N}))| < \cdots$). This is not surprising given that we can't even define infinite in first order. Our interest in the theorem is partly the role it plays in the Lindström theorems (see Flum (1975)) but also its utility in showing directly that some quite natural English constructions imply cardinal comparison and thus are not first order. The following, surprising, observation is due to Boolos (1981):

Theorem 9.7. *For A, B one place predicates, the sentence* "For every A there is a B" *is not expressible in first order logic.*

An illustrative example from Boolos is *For every philosopher that has studied Spinoza thoroughly, there is one that hasn't even read the Ethics.* A catchy example (which Boolos credits to E. Fisher) is *For every drop of rain that falls a flower grows.* Boolos points out that *For every A there is a B* is not correctly represented by the first order formula $\forall x(Ax \rightarrow \exists yBy)$ which just says (on reflection) that if there is an A then there is a B. It is logically equivalent to $(\exists xAx \rightarrow \exists yBy)$. Moreover we understand the association that *For every A there is a B* enforces between the As and the Bs to be one to one. Different drops of rain correspond to different flowers (though there may be some flowers unrelated to any drop of rain). Thus in effect *For every A there is a B* says that there is a one to one map from the set of As into the set of Bs, that is, $|A| \leq |B|$. And the direct proof of Theorem 9.7 is as follows.

Proof. Let $\varphi(n)$ be *There are at least n Bs* (which we have already seen how to represent in first order). Suppose, leading to a contradiction, that $\psi(A, B)$ is a first order sentence expressing that $|A| \leq |B|$, $-\psi(A, B)$ says that $|A| > |B|$. Set $K = \{\varphi(n) \mid n \in \mathbb{N}\} \cup \{-\psi(A, B)\}$. Clearly any finite subset K' of K has a model: for n the largest number such that $\varphi(n)$ is in K' choose B to be the n-membered set $\{1, 2, \ldots, n\}$ and choose $A = \{1, 2, \ldots, n, n+1\}$, so $|A| > |B|$. So by compactness K itself has a model. But $|K| = |\mathbb{N}|$ so by Löwenheim-Skolem K has a model with just $|\mathbb{N}|$ elements. And $|B| = |\mathbb{N}|$ since B is the set of natural numbers. So A is a subset of a set of cardinality $|\mathbb{N}|$ and so cannot have cardinality greater than $|\mathbb{N}| = |B|$, a contradiction. □

In fact the same technique here can be used to show that proportional *most* (in the sense of "more than half" is not first order. We note first that

(36) $\textsc{most}(A)(B) = T$ iff $|A \cap B| > |A - B|$, iff $\neg(|A - B| \leq |A \cap B|)$.

Now we go through Boolos' proof working with $\psi(A - B, A \cap B)$ instead of $\psi(A, B)$. Specifically, let $\varphi(n)$ be the first order formula expressing *There are at least n As that are not Bs.* For $n = 3$ for example $\varphi(n) = \exists x \exists y \exists z(x \neq y \,\&\, x \neq z \,\&\, y \neq z$

$$\& \, Ax \,\&\, Ay \,\&\, Az \,\&\, \neg Bx \,\&\, \neg By \,\&\, \neg Bz).$$

Then set $K = \{\varphi(n) \mid n \in \mathbb{N}\} \cup \{\neg\psi(A, B)\}$ where $\psi(A, B)$ says *Most As are Bs* and is assumed expressible in FOL (leading to a contradiction). To see that each finite subset of K has a model choose n maximal such that $\varphi(n) \in K$ and set $A = \{1, 2, \ldots, 2n + 1\}$ and set $B = \{n + 1, \ldots, 2n + 1\}$. Then B lacks the first n elements of A so $|\{A - B\}| = n$. $A \cap B$ contains just the remaining $n+1$ elements of A so $|A \cap B| = n+1 > |A - B|$. So *Most As are Bs* is true in this model. By compactness then K itself has a model. And since K is clearly denumerable K has a model of cardinality $|\mathbb{N}|$ by Löwenheim-Skolem. But K entails that for every n there are at least n elements of $A - B$, so that set has cardinality $|\mathbb{N}|$. But $A \cap B$ is a subset of the domain of cardinality $|\mathbb{N}|$ and so cannot have cardinality greater than $|\mathbb{N}|$ contradicting that $|A \cap B| > |A - B|$. Thus *Most As are Bs* is not expressible in first order after all.

This ends our excursus into the linguistic properties of FOL. Let us return now to the use of variable binding operators in representing English.

9.3 Extending the Class of VBO's: the Lambda Operator

The merits of FOL as a translation target for English have been amply attested in the literature. Its ability to represent binding and scope ambiguities has proven enlightening. But as a language in which to represent the meaning system of a natural language we want better, preserving its merits of course. On the one hand it seems unenlightening to destroy basic constituents of English structure. If phrases like *every student*, *some student*, etc. have no logical meaning why does English use them? If we needed to tear apart English DPs to interpret Ss containing them why don't we speak in a language closer to FOL to begin with? And second, we know that the expressive power of FOL is limited. It cannot express proportionality quantifiers or cardinality comparison. *I have more seashells than you do* is not a first order sentence.

So we return to $\mathcal{L}$(Eng), enriching it with the *lambda* operator λ yielding a language in which we can handle binding and scope ambiguities but retain the presence of DP constituents like *all/most/no students* which are directly semantically interpreted. We first enrich the syntax of our language, and then say how the new expressions are interpreted. Individual variables $x, y, z, \ldots$ are added in the category NP.

(37) λ abstraction (restricted)

syntax: If d is an expression of category C and x an individual variable then $\lambda x.d$ is an expression of category NP$\backslash C$. We also often write $\lambda x(d)$ for $\lambda x.d$.[3]

semantics: $[\![\lambda x.\varphi]\!]_{\mathcal{M}}$ maps each assignment α to a function with domain $E_{\mathcal{M}}$ whose value at each $b \in E_{\mathcal{M}}$ is $[\![\varphi]\!]_{\mathcal{M}}(\alpha^{x \to b})$.

So λx is a function creator: whatever type of object φ denotes, $\lambda x.\varphi$ denotes a function mapping entities to those objects (when $\lambda x.\varphi$ has no free variables—defined as before, but with the clause $\mathrm{FV}(\lambda x.\varphi) = \mathrm{FV}(\varphi) - \{x\}$ added). Later we remove the restriction that the lambda variable range just over the domain of the model. Thus, informally, in the language of arithmetic $\lambda x.(x > 5)$ denotes a function mapping a number n to T if $n > 5$ and to F otherwise. $\lambda x.(x$ CRITICIZE $x))$ maps *John* to T iff "x criticized x" is true when x is set to denote John. So it is true iff John criticized himself. Thus lambda allows us to bind variables without introducing any universal or existential commitment. Here are examples representing binding in $\mathcal{L}$(Eng) with the lambda operator.

(38) a. Every worker criticized himself.
a'. (EVERY WORKER)$\lambda x.(x$ CRITICIZE $x)$
b. Most students respect themselves.
b'. (MOST STUDENT)$\lambda x.(x$ RESPECT $x)$

In (38a') *x criticize x* has category P_0, so $\lambda x.x$ CRITICIZE x has category NP$\backslash$S $= \mathrm{P}_1$. *every worker* has category $\mathrm{P}_0/\mathrm{P}_1$ so it combines with it to yield (38a') of category P_0. And as $\lambda x.x$ CRITICIZE x denotes the property of criticizing oneself (e.g. it holds of John iff John criticized himself) our previous semantics tells us that (38a') is true iff WORKER $\subseteq \{b \in E \mid \text{CRITICIZE}(b)(b) = T\}$, that is, each worker is an object that criticized himself. Note too that (38a) and (38b) have the same syntactic form (except for plural marking in (38b)), and their logical representations are syntactically isomorphic as well. This isomorphism is maintained by the Ss resulting from replacing *every* in (38a) by other Dets, such as *some*, *no*, *most*, etc. So quantified DPs

[3] A slightly more accurate way of syntactically integrating the lambda notation into $\mathcal{L}$(Eng) is:

syntax: If d is an expression of category C and x a variable of category NP then $\lambda x.d$ is an expression of category NP$\backslash C$ if $C = S$ and of category C/NP otherwise.

In practice lambda expressions are primarily used in contexts where semantic interpretation is the issue so expressions are typically just written in function-argument order, as we will usually do.

are directly interpreted, not destroyed as in standard FOL translation semantics.

This approach also has the advantage that we can represent cases in which antecedents of anaphors, such as *every worker* in (38a) and *most students* in (38b), are not representable at all in first order, as is the case with *most students*.

Quantification and variable binding are independent operations.

Shortly we turn to some more complicated cases of binding and then of scope ambiguities, indicating the very significant utility of the lambda operator. But let's pause for a moment to wonder whether we have not lost something in eliminating the classical universal and existential quantifiers in favor of GQs. We have clearly gained something, the ability to represent binding by DPs like *most students*. But the classical semantics for universal and existential quantifiers made their boolean nature (glb, lub) apparent, whereas our set theoretic interpretations, repeated in (39), are not so directly boolean.

(39) a. $\text{EVERY}(A)(p) = T$ iff $A \subseteq \{b \in E \mid p(b) = T\}$
b. $\text{SOME}(A)(p) = T$ iff $A \cap \{b \in E \mid p(b) = T\} \neq \emptyset$

However these definitions have simpler equivalent formulations which are directly boolean (Keenan and Faltz (1985)) and indeed prove to be more useful than those above in certain contexts:

(40) a. $\text{EVERY}(A) = \bigwedge\{I_b \mid b \in A\}$
b. $\text{SOME}(A) = \bigvee\{I_b \mid b \in A\}$

Recall that I_bs are individuals—denotations of proper nouns, like *John*, and are defined by:

(41) For all properties $p \in [E \to \{T, F\}]$, all $b \in E$, $I_b(p) =_{df} p(b)$.

So in a given model, $\text{EVERY}(A)$ in (40a) would denote the same function as *John and Mary and Sue...* where the proper names run through the individuals in A. And the definition in (40a) yields the same result when applied to a property p as our earlier set theoretic definition:

(42) $\text{EVERY}(A)(p)$

$= (\bigwedge\{I_b \mid b \in A\})(p)$	Def EVERY in (40a)
$= \bigwedge\{I_b(p) \mid b \in A\}$	Pointwise meets
$= \bigwedge\{p(b) \mid b \in A\}$	Def individual
$= T$ iff for every $b \in A, p(b) = T$	Def glb in { T,F }
$= T$ iff $A \subseteq \{b \in E \mid p(b) = T\}$	Set Theory

The last line above is exactly the truth conditions for $\text{EVERY}(A)(p)$ on our earlier definition in (39a). And in a similar way (40b) shows that *some A* denotes the same as *John or Mary or Sue...* where the proper names run through the individuals with property A. Thus existential DPs behave denotationally like arbitrary disjunctions.

Exercise 9.9. Fill in the lines below proving that our two definitions of SOME are equivalent.

$$\begin{aligned}\text{SOME}(A)(p) &= (\bigvee\{I_b \mid b \in A\})(p)\\ &=\\ &\vdots\\ &= T \text{ iff } A \cap \{b \in E \mid p(b) = T\} \neq \emptyset\end{aligned}$$

We turn now to some (slightly) more complicated cases of binding. Observe that (43a) is ambiguous according as Bill is a threat to himself, (43b), or John is a threat to Bill, (43c).

(43) a. John protected Bill from himself.
b. $\lambda y.$(JOHN PROTECTED y FROM y)(BILL)
c. $\lambda y.$(y PROTECTED BILL FROM y)(JOHN)

Exercise 9.10. On the informal pattern of (43) use lambda binding to represent the binding in each of the Ss below.

a. Some doctor protected himself from himself.
b. Some doctor didn't protect himself from himself.
c. Some woman protected every patient from himself.
d. John protected himself from himself and so did Bill.

Exercise 9.11. Using lambda represent the two readings of the S displayed below. On one *his* is bound to *every child* and on the other it isn't bound at all.

Every child loves his mother.

We note that lambda can be used to bind long distance (across clause boundaries) as well:

(44) a. Each student thinks that every teacher likes him.
b. (EACH STDNT)($\lambda x.$(x THINKS EVERY TEACHER LIKES x))

We turn now to the use of lambda in representing scope ambiguities, as in (45a). Recall that our mode of direct interpretation (Chapter 7) with DPs interpreted in the position in which they occur in natural English captured the ONS (Object Narrow Scope) reading directly. But we didn't provide a way to represent the OWS (Object Wide Scope) reading. Now we can.

(45) a. Some student praised every teacher.
b. OWS: (EVERY TEACHER)($\lambda x.$(SOME STUDENT PRAISED x))

9.3.1 Generalizing the use of the lambda operator

Our examples so far have just combined λx with P_0s (formulas) to form a P_1. But we find instances of binding and scope ambiguities in expressions of other categories. For example, let us add nominals like *friend of*, *colleague of*, etc. to $\mathcal{L}$(Eng) as expressions that combine with a DP on the right to form an N, as in (46a).

(46) a. [SOME[FRIEND-OF[EVERY SENATOR]]]
b. (EVERY SENATOR)(λx.SOME FRIEND OF x)

In (46a) FRIEND OF EVERY SENATOR denotes a property, one that someone has iff he is a friend of every senator. And as usual with existentially quantified expressions, (46a) denotes the least upper bound of the individuals with that property.

But the wide scope reading of *every senator* in (46b) might be read as *for every senator, some friend of his.* It denotes the greatest lower bound of *some friend of* x_1, *some friend of* $x_2, \ldots$ where the x_i run through the individuals with the senator property. The interesting fact here is that the ambiguity seems local to the DP, rather than being an essential S-level ambiguity.

More widely used in linguistic description are cases where lambda binds a variable of type other than the type of individuals. This allows for more complicated lambda expressions than we have used so far, so it will be useful to introduce a widely used *type* notation which helps us check both the well formedness of lambda expressions and their semantic interpretation. Type will be a set whose elements are used to index expressions in such as way as to identify the set in which they denote (in a given model). Here first is a standard, minimal, set of types.

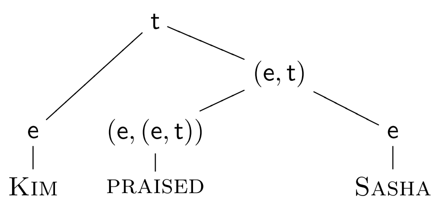
Definition 9.10. Type is the least set satisfying:

a. $\mathsf{e} \in$ Type and $\mathsf{t} \in$ Type, and
b. if a, b are both in Type then $(a, b) \in$ Type.

An expression of Type t will denote in the set $\{T, F\}$ of truth values, noted $\text{DEN}_E(\mathsf{t}) = \{T, F\}$. An expression of Type e will denote in the domain E of a model, so $\text{DEN}_E(\mathsf{e}) = E$. So the primitive Types correspond to the semantic primitives of the language. And in general an expression of Type (a, b) will denote a function from the denotations of Type a to those of Type b. That is, $\text{DEN}_E(a, b) = [\text{DEN}_E(a) \rightarrow \text{DEN}_E(b)]$. For example, an expression of Type (e, t) is just a property, a map from E into $\{T, F\}$. Common nouns like *doctor* and P_1s like *laugh* have this Type in our treatment (which has not represented tense which usually shows up on P_1s but not on Ns). An expression of Type $(\mathsf{e}, (\mathsf{e}, \mathsf{t}))$

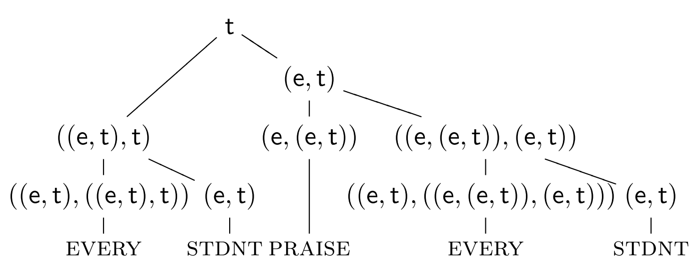
denotes a binary relation, one of Type $((\mathsf{e},\mathsf{t}),\mathsf{t})$ the Type of subject DPs, such as *he* and *she*, and $((\mathsf{e},(\mathsf{e},\mathsf{t})),(\mathsf{e},\mathsf{t}))$ the Type of object DPs, such as *him*, *her*, *himself*, and *herself*. Here is a "Type tree" for the P_0 *Kim praised Sasha* from $\mathcal{L}$(Eng) assuming *Kim* and *Sasha* are NPs (not DPs) here.

(47)



Note that assigning a type to DPs treated polymorphically as we have done appears problematic, with our minimal set Type, since as subjects DPs map P_1s to P_0s and should have type $((\mathsf{e},\mathsf{t}),\mathsf{t})$ like *he* and *she*, but as grammatical objects they map P_2s to P_1s and should thus have type $((\mathsf{e},(\mathsf{e},\mathsf{t})),(\mathsf{e},\mathsf{t}))$, like *him* and *himself*. So we shall extend the (standard) type notation above by adding (p_{n+1},p_n), where we think of p_{n+1} and p_n as variables ranging over the types for P_{n+1}s and P_ns respectively. We take its denotation set in a model $\mathcal{M}$ to be the maps from the union of P_{n+1}s into the union of the P_ns given in Definition 7.20 in Chapter 7.

(48)



So here the two occurrences of *every* and the two of *every student* have different types, reflecting in fact their different interpretations.

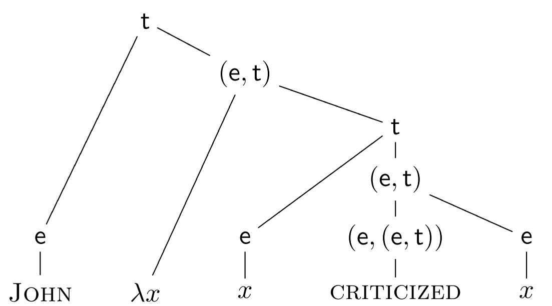
Exercise 9.12.

a. What type would you assign to manner adverbs (*slowly*, *gleefully*, etc.) given that they combine with P_1s to form P_1s?
b. Exhibit a plausible type tree for *Ruth talks rapidly.*
c. Exhibit a type tree for the P_0 *every student criticized himself.*
d. Assume you can only coordinate expressions of like category. Make a sensible guess at a type tree for *John criticized both himself and the teacher.*

e. How might we assign types to Prepositions (*to*, *for*, *from*, *with*,...) so that they combine with DPs to form predicate modifiers such as *with every teacher* in *Kim spoke with every teacher*. Using your type assignment exhibit a type tree for the (i), (ii) and (iii) below, and then say why we have no type for (iv).
   i. Kim spoke with every teacher.
   ii. Kim spoke with him.
   iii. Kim talks to himself.
   iv. *Kim talks to he.

We have so far used the lambda operator to bind variables of type e in expressions of type t. Here is a simple illustrative type tree using the lambda operator in this way:

(49)



In (49) λx combined with an expression x *criticized* x of type t to form an expression of type (e,t). We now generalize the use of lambda so that it can bind variables of any type σ in expressions of any type τ, forming expressions of type (σ, τ).

Exercise 9.13. Consider the use of the lambda operator in the P_0 below. What is the type of the expression it combines with and what type does it build?

[Mary[Dana[λx[PROTECTED x FROM x]]]]

The more general use of the lambda operator allows variables of all types. To know what type is intended we either say it explicitly in the text or we subscript the variable with its type. Thus the property of criticizing oneself which we represented above as $\lambda x(x$ CRITICIZED $x)$ would now be represented as $\lambda x_e(x$ CRITICIZED $x)$. It has type (e,t). In general if x is a variable of type σ and d an expression of type τ then $\lambda x.d$ has type (σ,τ). For the formal record,

Definition 9.11 (Lambda Extraction (unrestricted))**.**

a. **Syntax:** For x a variable of type σ and d an expression of type τ

$\lambda x.d$ is of type (σ, τ).

b. **Semantics:** For each model $\mathcal{M}$, each assignment α and each $b \in \text{DEN}_E(\sigma)$, $[\![\lambda x.d]\!]_{\mathcal{M}}(\alpha)(b) = [\![d]\!]_{\mathcal{M}}(\alpha^{x \to b})$.

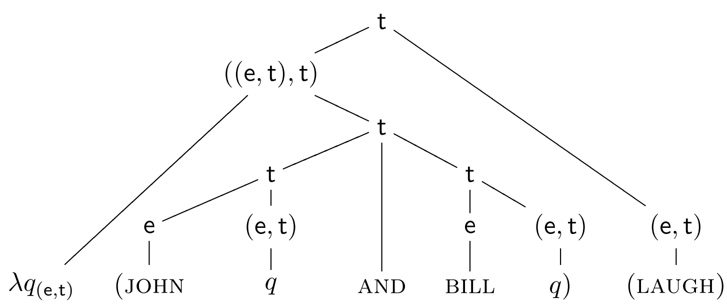
The definition in 9.11 assumes that we have denumerably many variables of all types and that an assignment α maps each variable of each type to the denotation set associated with that type.

Now English presents a variety of expressions in which we bind expressions of type other than e, so the more general use of lambda is enlightening. Consider for example the use of the "pro-verb" *so do* in (50a). We might represent this binding as in (50b)

(50) a. John laughed and so did Bill.
b. $\lambda q_{(\mathsf{e},\mathsf{t})}(\text{JOHN } q \text{ AND BILL } q)(\text{LAUGH})$

To verify that (50b) is of type t and thus a truth value denoting expression, here is its type tree:

(51)



As lambda expressions become more complicated sketching their type tree is a useful way to verify that our expressions are well formed and have the type we want them to. For example in the next exercise some of the sisters to lambda expressions are themselves lambda expressions.

Exercise 9.14. Exhibit lambda representations for each of the following. Give two representations in cases where the expression is marked as ambiguous:

a. John didn't laugh and neither did Bill.
b. John criticized himself and so did Bill.
c. No philosopher thinks he's clever and Bill doesn't either. (not ambiguous with *he* bound to *no philosopher*)
d. John thinks he's clever and so does Bill. (Two readings, with *he* bound in both)
e. The woman that every Englishman likes best is his mother. (with *his* bound to *every Englishman*)

Simplifying lambda expressions

The increasing complexity of lambda expressions can be lessened by various reduction operations. Here are three widely used techniques.

First, it is often useful to "rename variables", that is, to replace a lambda expression with an alphabetic variant. This is sometimes referred to as *α-conversion*, and is occasionally necessary when applying other reduction mechanisms discussed below.

Second, if we syntactically combine a lambda expression $\lambda x.\varphi$ with an expression b of the same type as the variable x introduced by λ and usually written after the lambda expression $(\lambda x.\varphi)(b)$, our semantics tells us that, with one restriction (Trap 2 below), this expression is logically equivalent to the one we get by replacing all free occurrences of x in φ with b, noted $\varphi[x\backslash b]$. For example:

(52) (λx.MARY PROTECTED x FROM x'S FATHER)(b)
$\equiv$ (MARY PROTECTED x FROM x'S FATHER)$[x\backslash b]$
= MARY PROTECTED b FROM b'S FATHER.

This process of substitution is called *β-conversion* (or *β-contraction*) here noted $\Rightarrow$.

Exercise 9.15. For each expression below give the step by step result of applying β-conversion. The first case is done to illustrate the step by step process. The last two examples involve vacuous binding.

a. $(\lambda x.(\lambda y.[[\textsc{love}\ x]y])(j))(m)$
$\Rightarrow (\lambda y.[[\textsc{love}\ m]y])(j)$
$\Rightarrow [[\textsc{love}\ m]j]$
("john loves mary")
b. $(\lambda y.(\lambda x.[[\textsc{love}\ x]y])(j))(m) \Rightarrow$
c. $(\lambda x.[\textsc{sleep}\ (j)])(m) \Rightarrow$
d. $(\lambda x.(\lambda x.[[\textsc{love}\ x]x])(j))(m) \Rightarrow$

In general if x is not among the free variables of φ then $\lambda x.\varphi(b) \Rightarrow \varphi$.

In reasoning with complex expressions using many lambdas the use of β-conversion may be helpful in reducing their level of variable binding complexity (but not always). But in using β-conversion there are two traps you must be aware of:

Trap 1. β-conversion can not apply when the variable and the replacing expression have different types. For example in (53a) we cannot replace the x's, of type e, with *some student*, taken here for illustrative purposes to be of type $((\mathsf{e},\mathsf{t}),\mathsf{t})$. If we do substitute we obtain (53b) which does not have the same meaning as (53a), and we intend of course that β-conversion preserve meaning..

(53) a. $(\textsc{some student})\lambda x_{\mathsf{e}}(x \textsc{ laughed and } x \textsc{ cried})$
b. *(some student laughed and some student cried)

Trap 2. The expression we replace the variable with must not itself contain free variables that would, after the substitution, occur in the scope of a VBO already present in the host expression. For example, using variables p of type t, (54a) β-converts to (54b), but we cannot apply β-conversion to (55a), as the variable x in 'dance(x)' would become bound by a quantifier already present in the host expression.

(54) a. $(\lambda p_{\mathsf{t}}(\exists x(\textsc{sang}(x) \,\&\, p)))(\textsc{danced}(y))$
b. $\Rightarrow \exists x(\textsc{sang}(x) \,\&\, \textsc{danced}(y))$

(55) a. $(\lambda p_{\mathsf{t}}(\exists x(\textsc{sang}(x) \,\&\, p)))(\textsc{danced}(x))$
b. $\not\Rightarrow \exists x(\textsc{sang}(x) \,\&\, \textsc{danced}(x))$

Some authors (see Hindley and Seldin (1986)) allow conversion in cases like (54) building in that we trade in the host expression for an alphabetic variant with a totally new variable thereby avoiding the unintentional binding problem. We can always do this since we have infinitely many variables and each formula only uses finitely many. For example if we trade in the host expression $(\lambda p_{\mathsf{t}}(\exists x(\textsc{sang}(x) \,\&\, p)))$ in (54a) for $(\lambda p_{\mathsf{t}}(\exists z(\textsc{sang}(z) \,\&\, p)))$ then we can correctly derive the β-reduced form $\exists z(\textsc{sang}(z) \,\&\, \textsc{danced}(x))$.

Exercise 9.16. Represent each of the expressions using lambda binding and then present the result of applying β-conversion. I illustrate with the first example.

a. John said he would pass the exam, and pass the exam he did (Topicalization)
$\lambda x.((x \textsc{ say } x \textsc{ pass the exam})\,,\&\,(\lambda p_{(e,t)}(p(x))(\textsc{pass the exam}))(j)$
$\Rightarrow j$ say j pass the exam and $(\lambda p_{(e,t)}(p(j))(\textsc{pass the exam})$
$\Rightarrow j$ say j pass the exam and j pass the exam
b. John said he would punish himself and punish himself he did
c. Bill I really like but Fred I don't
d. What John is is proud of himself (Pseudocleft)
e. John bought and Bill cooked the turkey (RNR)
f. John interviewed the President and Bill the Vice President (Gapping)

Exercise 9.17. Which of the following expressions can be β-converted? In each case if your answer is "yes" exhibit the result of β-conversion. If your answer is "no", say why conversion does not apply. Then perform a change of variables so that conversion does apply and exhibit the result. Assume w, x, y, and z are all variables of type e, *love* is of type $(\mathsf{e},(\mathsf{e},\mathsf{t}))$.

a. $(\lambda x.(\lambda y.[[\text{LOVE } x]y])(z))(w)$
b. $(\lambda x.(\lambda y.[[\text{LOVE } x]y])(x))(y)$
c. $(\lambda x.(\lambda y.[[\text{LOVE } x]y])(y))(x)$

Exercise 9.18.

a. Draw the type tree for (i) below. It's root node should be (e,t). Then apply β-conversion and draw the type tree for the result. It's root node should also be (e,t).
   i. $\lambda F_{(\mathsf{e},\mathsf{t})}(\lambda x_{\mathsf{e}}(Fx))(G(e,(e,t))(y_e))$
   ii. $\Rightarrow$

b. Can we apply β-conversion to iii. below? If we do what result do we get? That result does have the right type, (e,t), but does not denote the same function as the expression in ii.
   iii. $\lambda F_{(\mathsf{e},\mathsf{t})}(\lambda x_{\mathsf{e}}(Fx))(G(e,(e,t))(x_{\mathsf{e}}))$

A last reduction process, at times quite useful, is:

$$\lambda x_a.F_{(a,b)}x \Rightarrow F_{(a,b)}.$$

This reduction is reasonable since the value of the function denoted by $\lambda x_a.F_{(a,b)}x$ at an argument d is the same as that denoted by F at d, whence $\lambda x_a.F_{(a,b)}x$ and F are the same function.

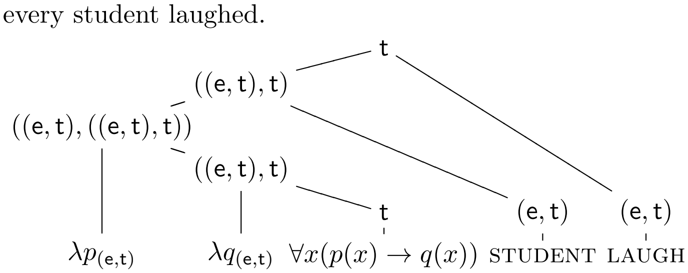
λ Definitions of Quantifiers

We have been treating Dets, such as *every*, *some*, *no*, and *most* as having type ((e,t),((e,t),t)). (We just need to define them in this type, as we have already said how the interpretation is extended to maps from $n+1$-ary relations to n-ary ones, $n > 0$). And for Dets which are first order definable we can find a first order formula such that by lambda abstracting twice over property denoting variables will indeed yield an expression of type ((e,t),((e,t),t)). For example we might define *every* of this type by:

(56) every $=_{df}$ $\lambda p_{(\mathsf{e},\mathsf{t})}(\lambda q_{(\mathsf{e},\mathsf{t})}(\forall x_{\mathsf{e}}(p(x) \to q(x)))$

Using the definition in (56) the type tree for (57a) can be given as in (57b):

(57) a. every student laughed.
b.



Exercise 9.19. On the pattern in (56) define:

a. some
b. no
c. exactly one

We illustrate this way of representing quantifiers as the student may encounter it in the literature, but we will not use it in what follows since our set theoretical definitions are both notationally much simpler and, crucially, more general. For example they treat proportionality quantifiers such as *most* comparably to *all*, *some*, and *no* despite not being first order definable.

9.3.2 Concluding reflection on scope and compositionality

The lambda operator gives us a means of representing the object wide scope (OWS) reading of sentences like (58a), again using upper case for denotations in a model.

(58)
a. Every student knows some teacher.
b. ONS reading:
$(\text{EVERY}(\text{STUDENT}))(\text{SOME}(\text{TEACHER})(\text{KNOW}))$
c. OWS reading:
$(\text{SOME}(\text{TEACHER}))\lambda x.((\text{EVERY}(\text{STUDENT}))(\text{KNOW}(x)))$

Naively the ONS reading in (58b) is natural in that it just interprets constituents of the expression is interprets. In particular *some teacher* is assigned a meaning, and so is *knows some teacher*. But in (58c) in order to get *some teacher* interpreted with scope over *every student* we have destroyed the P_1 constituent *knows some teacher* by lambda abstracting over the object of *know* and forming an (e,t) type expression, namely $\lambda x.((\text{EVERY}(\text{STUDENT}))(\text{KNOW}(x)))$. This is unnatural as we trade in the expression we are interpreting (58a) for a different one, one with different c-command relations. In effect, we are interpreting *every student knows* as a property, one that holds of an entity b just in case $(\text{EVERY}(\text{STUDENT}))(\text{KNOW}(b)) = T$. And to do this we have introduced the expression λ and two occurrences of the variable x, none of which have an obvious exponent in English.

Are we free to lambda abstract freely when interpreting English expressions? With no constraints this would make expressions with many DPs massively ambiguous. A sentence like *Some dean thought every student gave each teacher two apples* would be 24 ways ambiguous according to which of the four DPs had widest scope, then which of the three remaining had next widest scope, etc. In general an expression with n quantified DPs would be $n!$ ways ambiguous if all scope interpretations are possible. But while scope ambiguities do exist, as in the examples we have cited earlier, it does seem that in English Ss of the form Subject+Verb+Object, there is an interpretative asymmetry: the ONS reading is essentially always available and the object OWS reading only available in limited cases. For example competent speakers of English do not assert (59a,b) intending the OWS reading roughly paraphrased in (59a',b').

(59) a. No student answered every question correctly.
a'. *OWS: Every question has the property that no student answered it correctly.
b. Each student answered no question correctly.
b'. *OWS: No question has the property that each student answered it correctly

Just how best to characterize object wide scope readings is, at time of writing, still an actively researched question. One additional possibility we offer for consideration here is function composition. Recall that when F and G are functions with the range of G included in the domain of F then $F \circ G$ is that function with domain G and codomain that of F, given by: $(F \circ G)(b) = F(G(b))$. We can think of the composition operator, $\circ$, as a storage operator. It holds first F, then also G in store until it gets an argument to which one may apply yielding a value that the other can apply to. Compare the interpretations (in upper case) of (60) and (63):

(60) most male doctors = (MOST(MALE(DOCTOR)))

The phrase *most male* not a constituent of (60) and is not interpreted. Generally complex constituents C are interpreted by Function Application—one of Cs immediate constituents denotes a function taking the denotations of the other(s) as argument(s) and the interpretation of C is the value of that function at those arguments. Summarizing, let us adopt this as a principle:

(61) **Interpretive Principle 1 (IP1):** Branching constituents, and only branching constituents, are interpreted by Function Application.

Now let us consider a second tier interpretative mechanism, one that can only apply (and then not always) when IP1 cannot:

(62) **Interpretative Principle 2 (IP2):** Adjacent expressions which are not sisters may be interpreted by function composition when their domains and ranges allow it [4].

[4] Refinement of IP2 is needed for the (seemingly) rare cases where adjacent non-sisters denote functions that may compose in either order but with different mean-

Note that IP2 provides a second way of interpreting (60) since *most* and *male* are adjacent but are not sisters, and MOST and MALE may compose as (MOST $\circ$ MALE). Abbreviating (e,t) by p, MALE has type (p,p) it maps properties (DOCTOR) to properties (MALE(DOCTOR)). MOSThas type (p,(p,t)), so MOST $\circ$ MALE has type (p,(p,t)) and

(63) (MOST $\circ$ MALE)(DOCTOR) = MOST(MALE(DOCTOR)).

So interpreting *most male doctors* this way has the same value as in (60), so it buys us nothing. But consider the more enlightening case in (64).

(64) a. Most male and all female doctors (objected to that).
b. ((MOST $\circ$ MALE) $\wedge$ (ALL $\circ$ FEMALE))(DOCTOR)
= (MOST $\circ$ MALE)(DOCTOR) $\wedge$ (ALL $\circ$ FEMALE)(DOCTOR) Pointwise $\wedge$
= MOST(MALE(DOCTOR)) $\wedge$ (ALL(FEMALE))(DOCTOR) Def $\circ$

In (64a) *most* and *male* are adjacent and do not form a constituent and since the range of MALEis the same as the domain of MOST, properties in both cases, we may compose them in the order given. Similarly for *all* and *female*. And since MOST $\circ$ MALE and ALL $\circ$ FEMALE lie in the same boolean type, (p,(p,t)), we can take their greatest lower bound (the first line in (64b), which provably behaves pointwise, justifying the second line in (64b). So IP2 allows us to interpret the non-constituent coordination in (64a) without recourse to lambda abstraction or syntactic distortion. (Probably most linguists would think of deriving (64a) syntactically from *most male doctors and all female doctors* by some process of Right Node Raising (RNR) and, the claim would go, it is this "underlying" expression which is compositionally interpreted. The

ings. One such case are certain adjectives, such as *small* and *expensive*. Both are restricting: a small house is a house, as is an expensive house. Treating house as of type (e,t), we may treat adjectives as of type ((e,t),(e,t)). That means in principle that they can semantically compose. Often only one order is natural: *a Russian medical text* is a fine doubly modified expression, whereas * *? a medical Russian text* is not. But *small* and *expensive* Keenan and Faltz (1985) are reasonably natural in either order, and the different composition orders yield different results:

a. a small [expensive house] is small relative to expensive houses
b. an expensive [small house] is costly relative to small houses

It might be in some state of affairs that the smallest of the expensive houses were large compared to houses in general, so none was a small house, not even an expensive small house. So

$$(\text{SMALL} \circ \text{EXPENSIVE})(\text{HOUSE}) \neq (\text{EXPENSIVE} \circ \text{SMALL})(\text{HOUSE}).$$

In this case the choice between these two interpretations is the one in which the initial function in the composition is the one denoted by the leftmost adjective in the expression.

distinct audible one in (64a) inherits that interpretation. This approach does depend on an undefined syntactic operation (RNR). This definition is quite non-trivial; for example, it cannot derive (65a) from (65b) preserving meaning.

(65) a. most male and all female doctors at some Midwestern university
b. most male doctors at some Midwestern university and all female doctors at some Midwestern university

The composition approach involves fewer complications (lambda abstraction, RNR) and just interprets what is said. It thus yields a more direct account of why we speak the way we do rather than just using the underlying sources for expressions like (64a) and (65a). This approach to natural language interpretation enforces a kind of truth-in-advertising:

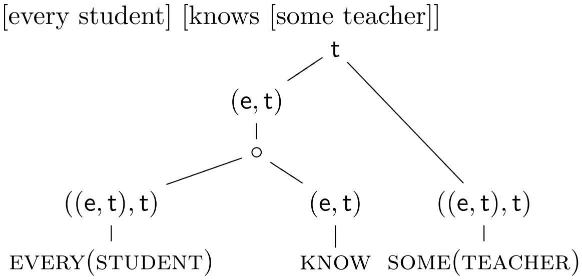
What you say is what you mean.

Cooper (1982) refers to this guideline as *Wholewheat syntax (unenriched with inaudibilia).*

Exercise 9.20. Illustrate how to interpret *Bill I like* without lambda abstraction, just function composition and function application.

Returning now to object wide scope readings, we see that we can insightfully use function composition here as well. Taking *every editor* to be of the lowest type it can have, ((e,t),t) and *read* to be of type (e,(e,t)) we see (66a,b) that EVERY(STUDENT) composes with KNOW to yield a function of type (e,t), given below:

(66) a. [every student] [knows [some teacher]]
b.



In (66a) *every student* asymmetrically c-commands *some teacher*. But in the type tree (66b) for its semantic interpretation, we have interpreted the adjacent expressions *every student* and *know* by function composition (they do not form a constituent and are not sisters). So in the type tree, SOME(TEACHER) asymmetrically c-commands EVERY(STUDENT). So interpreting adjacent expressions by function composition allows a syntactically c-commanded DP to take semantic scope over a c-commanding one. Further SOME(TEACHER) is a sister to the property (EVERY(STUDENT))$\circ$KNOW and by Function Application maps it to a truth value. Consider now just what property is denoted by ((EVERY(STUDENT)) $\circ$ KNOW. Its value at an entity b is given by:

(67) $((\text{EVERY}(\text{STUDENT})) \circ \text{KNOW}))(b)$
$= (\text{EVERY}(\text{STUDENT}))(\text{KNOW}(b))$ Def $\circ$
$= (\lambda x.((\text{EVERY}(\text{STUDENT}))(\text{KNOW}(x)))(b)$ β-conversion

In other words, the composition of EVERY(STUDENT) with KNOW is the same function as that obtained by lambda abstraction in

$$\lambda x.((\text{EVERY}(\text{STUDENT}))(\text{KNOW}(x))).$$

Thus in this case we obtain the same semantic result but without using the lambda operator or bound variables. Once again Occam's Razor cuts our way. To summarize the differences in the two approaches:

(68)
a. The lambda abstraction approach gets the object wide scope reading of Ss like (58a) by assigning the English expression two different syntactic representations, each one compositionally interpreted (just by Function Application).
b. The function composition approach does not change the logical syntax of the expression—it only interprets what we see, but it is non-compositional in allowing non-constituents to be interpreted (just by function composition).

Exercise 9.21. Provide a type tree just using function composition and function application for

(69) John bought and Bill cooked the turkey.

To conclude, we are not claiming that function composition is the best way to represent scope ambiguities. We are only illustrating the use of a mathematical tool which seems enlightening in some cases. Its range of uses is a matter of further empirical investigation. Here is a last suggestive case different in character from the syntactically oriented ones above.

Several languages (German, French, Spanish, Italian, Greek, Hebrew) admit phonological words which are naturally interpreted as the composition of a Preposition denotation with a definite article (*the*) denotation. (70) from German illustrates the, roughly, optional contraction of *an* 'on' + *dem* (the: neuter sg.) to *am*:

(70) *Ich habe mich* { *an dem* / *am* } *Regal*
I have me { on the:dat / on+the } bookshelf
verletzt
hurt

'I hurt myself on the bookshelf.'

German admits of about ten such combined forms. Ones like *am* above can perhaps be treated as optional "contractions", the interpreted form just being the uncontracted *an+dem*. So to interpret the S with *am* we would modify its syntax, replacing *am* with *an dem*. However, we could directly interpret *am* by listing it as a lexical item and specifying its semantic interpretation as AN ∘ DEM. The latter approach seems more strongly motivated when the "contraction" is obligatory, as is the case with such forms in French and Hebrew. Thus in French *á + le* 'to the:m.sg' is obligatorily realized as *au* (/o/). And 'from the:m.sg' is obligatorily *du* rather than *de+le*. In contrast no short form is available with the feminine sg. article *la*, as we see in (71a).

(71) a. *Je rentre á la maison á midi*
I return to the:fem.sg house at noon

'I return home at noon.'

b. *Le bureau est fermé le soir*
The:masc.sg office is closed the evening

'The office is closed in the evening.'

c. *Je vais au (*á le) bureau á huit*
I go to+the:m.sg (*to the:m.sg) office at eight
heures le matin
hours the morning

'I go to the office at 8 o'clock in the morning.'

Here it would seem a simple matter to enter *au* and *du* in the French lexicon with the meanings A∘LE and DE∘LE. In a similar way in Hebrew 'to the' is obligatorily *la*, whose meaning is Lə ∘ HA, the composition of the denotation of the goal locative *lə* with that of the definite article *ha*. Similarly 'at + the' is obligatorily *ba* = Bə ∘ HA. (72a,b,c) illustrate the goal locative case.

(72) a. *ha-more moxer sfarim lə-studentim*
the-teacher sells books to+students

'The teacher sells books to students.'

b. *ha-studentim lo baim mukdam*
the-students not come early

'The students don't come early.'

c. *ha-more* *moxer* *sfarim* *la-studentim*
the-teacher sells books to+the-students
'The teacher sells books to the students.'

To conclude, we note that the linguistic literature concerning scope ambiguities, binding and compositional interpretation is massive, well beyond the scope of a text such as this. For a sample of approaches we refer the reader to Heim and Kratzer (1998), Szabolcsi (1997), Barker and Jacobson (2007), Moortgat (1996), and Steedman and Baldridge (2007).

10

Semantics IV: DPs, Monotonicity and Semantic Generalizations

In this chapter we focus on semantically based generalizations concerning English DPs. An important, even tantalizing, role is played by *monotonicity* properties.

10.1 Negative Polarity Items

We begin with a classical observation in generative grammar which concerns the distribution of *negative polarity items* (*npi*'s) like *ever* and *any*. Klima (1964) observed that npi's do not occur freely but require negative contexts, such as *n't* or *not*, as in:

(1) a. John has*n't ever* been to Pinsk.
    b. *John has *ever* been to Pinsk.

(2) a. John did*n't* see *any* birds on the walk.
    b. *John saw *any* birds on the walk.

However, certain DPs in subject position also license npi's:

(3) a. No student here has *ever* been to Moscow.
    b. *Some student here has *ever* been to Moscow.

(4) a. Neither John nor Mary know *any* Russian.
    b. *Either John or Mary know *any* Russian.

(5) a. Neither student answered *any* question correctly.
    b. *Either student answered *any* question correctly.

(6) a. None of John's students has *ever* been to Moscow.
    b. *One of John's students has *ever* been to Moscow.

The *a*-expressions are grammatical, the *b*-ones are not. But the pairs just differ with respect to their initial DPs, not the presence vs. absence of *n't* or *not*.

Two linguistic problems:

1. Define the class of DPs which license npi's, as above, and
2. State what, if anything, the licensing DPs have in common with *n't/not*.

A syntactic attempt (see for example Linebarger (1987)) to solve both problems would be to say that just as *n't* is a reduced form of *not*, so *neither...nor...* is a reduced form of [*not (either...or...)*], *none* is a reduction of *not one*, and *no* of *not a*. The presence of n in the reduced forms is explained as a remnant of the original *not*[1]. So on this view the licensing DPs above "really" contain a *not*, and that is what they have in common with *n't*. Note in support of this claim that DPs overtly built from *not* do license npi's:

(7) a. Not a single student here has *ever* been to Moscow.
  b. Not more than five students here have *ever* been to Moscow.

But this solution seems insufficiently general (Ladusaw (1983)): The initial DPs in the *a*-Ss below license npi's; those in the *b*-Ss do not. But neither contains an *n*-word.

(8) a. Fewer than five students here have *ever* been to Moscow.
  b. *More than five students here have *ever* been to Moscow.

(9) a. At most four students here have *ever* been to Moscow.
  b. *At least four students here have *ever* been to Moscow.

(10) a. Less than half the students here have *ever* been to Moscow.
  b. *More than half the students here have *ever* been to Moscow.

(11) a. At most 10% of the players here will *ever* get an athletic scholarship.
  b. *At least 10% of the players here will *ever* get an athletic scholarship.

Ladusaw supports a more comprehensive, semantic, answer: the licensing DPs are just the *downward entailing* ones, as defined informally below. First, the definition of *upward entailing*.

Definition 10.1. Let X be a DP. X is *upward entailing* iff the following argument type is valid (meaning the premises entail the conclusion):

Premise 1: All Ps are Qs
Premise 2: X is a P

[1] In fact this *n-* derives historically from Anglo-Saxon *ne* (ultimately Proto-Indo European *ne*). The initial *n* in *not* is this same *n*.

Conclusion: X is a Q

For example, Proper Names are upward entailing: if all Ps are Qs and Mary is a P we can infer that Mary is a Q. For example, if all students are vegetarians and Mary is a student then Mary is a vegetarian. (This is just an Aristotelian syllogism).

Note that if X is a plural DP then the appropriate form of Premise 2 above is "X are Ps", and the appropriate form of the conclusion is "X are Qs". For example, *at least two doctors* is upward entailing since if all Ps are Qs and at least two doctors are Ps then those two doctors are Qs, so we can infer that *at least two doctors* are Qs.

(12) Some upward entailing DPs:

a. he, she, John, Mary
b. DPs of the form [Det + N], where Det = some, more than ten, at least ten, most, at least/more than half the, infinitely many, all, every, several, at least/more than two out of three, at least/more than 10% of (the), at least/more than a third of (the), the ten, the ten or more, John's, John's ten (or more), both,
c. Partitive DPs like [Det of John's students], where Det is as in (b) above.

And here are two ways of building syntactically complex upward entailing DPs:

Facts:

1. If X and Y are upward entailing then so are (X and Y) and (X or Y). But (not X) and (neither X nor Y) are not upward entailing (when X and Y are non-trivial[2]).
2. If X is upward entailing then so are possessive DPs of the form [X's N].

From Fact 1 *either John or some official* is upward entailing since *John* is, and, by (12b), *some official* is. Fact 2 implies that *every player's agent* is upward entailing, since *every player* is by (12b). And this is correct: if all Ps are Qs and every player's agent is a P then every player's agent is a Q.

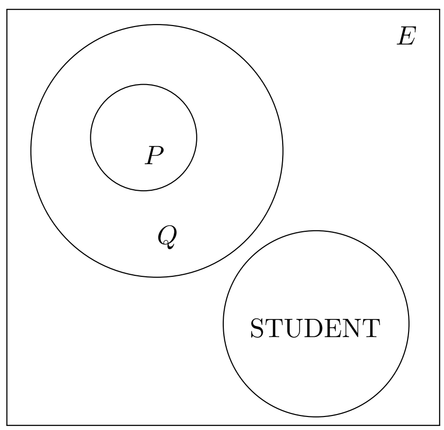
We now turn to downward entailing DPs.

[2]There are just two DP denotations which are both increasing and decreasing: $\mathbf{0}$, which maps all sets to F, and $\mathbf{1}$, which maps all sets to T. These are expressible by *fewer than zero* students and *either all students or else not all students* respectively. One computes $\neg\mathbf{0} = \mathbf{1}\neg\mathbf{1} = \mathbf{0}$, so the complements of these trivial denotations are themselves trivial and thus both increasing and decreasing.

Definition 10.2. Let X be a DP. X is *downward entailing* iff the following argument is valid:

Premise 1: All Ps are Qs
Premise 2: X is a Q
Conclusion: X is a P

For example, *no student* is downward entailing, since if all Ps are Qs and no student is a Q then no student is a P. For example, if all poets are vegetarians and no athlete is a vegetarian then indeed no athlete is a poet. The following Venn diagram is helpful here. It correctly reflects the set inclusion of Premise 1, and the fact that the student set does not intersect Q shows that no student is a Q. The inference that no student is a P is obvious, since if the student set intersected P it would have to intersect Q, and it doesn't.



(13) Some downward entailing DPs:
  a. Ones of the form [Det + N], where Det = no, neither, less/fewer than five, at most five, less than/at most half the, less than 10% of the/john's, less than one ...in ten,
  b. Partitive DPs, like [Det of John's students], where Det is as in (a) above.

And here are some ways of building complex downward entailing DPs:

Facts:

3. If X and Y are upward entailing then *neither* X *nor* Y is downward entailing.
4. If X is upward entailing then *not* X is downward entailing.
5. If X and Y are downward entailing then so are (X and Y) and (X or Y).

6. If X is downward entailing then so are possessive DPs of the form [X's N].

From Fact 3 we infer that *neither John nor Mary* is downward entailing since both *John* and *Mary* are upward entailing. And it is: if all Ps are Qs and neither John nor Mary is a Q then clearly neither can be a P. Fact 4 implies, correctly, that DPs like *not more than ten boys* and *not more than half the women* are downward entailing. Fact 5 tells us that *no student and not more than two teachers* is downward entailing. And finally the more interesting Fact 6 tells us that *no child's doctor* is downward entailing since *no child* is. And this is correct: if all Ps are Qs and no child's doctor is a Q then, clearly, no child's doctor can be a P. Specifically, if all poets are vegetarians and no child's doctor is a vegetarian then, indeed, no child's doctor is a poet.

Exercise 10.1. Show that Fact 6 entails that *no actor's agent's doctor* is downward entailing.

And now we can offer an answer to the first linguistic question above:

(14) The subject DPs that license npi's are just the downward entailing ones.

10.2 Monotonicity

(15) is a substantive semantic generalization. But it does not answer the second question: what do downward entailing DPs have in common with *n't*/*not*?

To provide an answer to this question we shall generalize the notions of upward and downward entailing so that they apply to categories of expressions other than DP. To this end recall that a poset is a pair $(A, \leq)$ with A a non-empty set and $\leq$ a reflexive, antisymmetric and transitive binary relation on A. We saw that $(\{T, F\}, \leq)$ is a poset, where $\leq$ is the implication order (for all $X, Y \in \{T, F\}$, $X \leq Y$ iff if $X = T$ then $Y = T$). And the set of possible P_1 denotations, $[E \rightarrow \{T, F\}]$ is a poset pointwise.

Now DPs denote Generalized Quantifiers (GQs), that is, functions from $[E \rightarrow \{T, F\}]$ into $\{T, F\}$. So GQs are functions from a poset to a poset. And we generalize:

Definition 10.3. Let $(A, \leq)$ and $(B, \leq)$ be posets. Let F be a function from A into B. Then

a. F is *increasing* iff for all $x, y \in A$, if $x \leq y$ then $F(x) \leq F(y)$. Increasing functions are also called *isotone* and are said to *respect the order*.

b. F is *decreasing* iff for all $x, y \in A$, if $x \leq y$ then $F(y) \leq F(x)$. Decreasing functions are also called *antitone*, and are said to *reverse the order*.

c. F is *monotone* iff F is increasing or F is decreasing.

Increasing functions are often, redundantly, called *monotone increasing*; decreasing ones *monotone decreasing*. (And occasionally *monotone* by itself is used just to mean monotone increasing).

Now the DPs we called upward entailing above are just those that denote increasing functions from $[E \to \{T, F\}]$ into $\{T, F\}$. Compare the two formulations:

(15) The DP *some student* is upward entailing:
Premise 1: All Ps are Qs
Premise 2: Some student is a P
Conclusion: Some student is a Q

(16) The GQ SOME STUDENT is increasing:
if $p \leq q$ then if (SOME STUDENT)$(p) = T$
then (SOME STUDENT)$(q) = T$.

(Recall that for p, q P_1 denotations, $p \leq q$ iff for all $b \in E$, if $p(b) = T$ then $q(b) = T$).

Clearly (15) and (16) say the same thing. "All Ps are Qs" just says that the denotation p of P is $\leq$ the denotation q of Q. And semantically the second line says that (SOME STUDENT)$(p) = T$. To conclude that (SOME STUDENT)$(q) = T$ just says then that SOME STUDENT is increasing. (15) and (16) just differ in that the former talks about expressions and the latter talks about their denotations. Similarly downward entailing DPs are just those that denote decreasing functions.

Exercise 10.2. Analogous to (15) and (16) above, show that the claim that *no student* is downward entailing is the same claim as that its denotation is decreasing.

We could now reformulate (14) to say that the DPs which license npi's are just the decreasing ones. But let us generalize a little further:

(17) The Ladusaw-Fauconnier Generalization (LFG)
Negative polarity items only occur within the arguments of decreasing expressions.

The LFG enables us to see what decreasing DPs have in common with negation: they both denote decreasing functions. Consider *n't*/*not* as a P_1 level function. Semantically it maps a property p to $\neg p$, that property which maps b to T iff p maps b to F. This function is a map from a poset $[E \to \{T, F\}]$ to a poset $[E \to \{T, F\}]$ and is easily seen to be decreasing:

Proposition 10.1. *For* $p, q \in [E \to \{T, F\}]$, *if* $p \leq q$ *then* $\neg q \leq \neg p$.

Proof. Assume the antecedent and let $(\neg q)(b) = T$. We must show that $(\neg p)(b) = T$. But since $(\neg q)(b) = T$ we infer that $q(b) = F$, whence $p(b) = F$ since $p \leq$ (if $p(b) = T$ then $q(b) = T$, a contradiction). And since $p(b) = F$ then $(\neg p)(b) = T$, as was to be shown. □

Recall that in any boolean lattice $x \leq y$ iff $\neg y \leq \neg x$, so this proposition is just a special case of a general boolean regularity. And the claim that negation is decreasing does not depend on taking it to be a P_1 level operator as long as it denotes the boolean complement function in the appropriate denotation set. For example, taking *not* to be a sentence level operator (as in classical logic, but not with much motivation in English) it would denote the truth functional $\neg$ which maps T to F and F to T. That $\neg$ is clearly decreasing.

Exercise 10.3.

a. Prove: For all $X, Y \in \{T, F\}$ if $X \leq Y$ then $\neg Y \leq \neg X$.
b. Prove: For an arbitrary boolean lattice B that for all $x, y \in B$, if $x \leq y$ then $\neg y \leq \neg x$.

Observe, (12b,c) and (13a,b), that for DPs of the form Det + N, the monotonicity of the DP is decided by the Det. But there are many monotone DPs that are not of the form Det + N. Names are increasing for example, *neither John nor Mary* is decreasing, and (where grammatical) negations of monotone DPs are monotone.

Exercise 10.4. In each case below say informally why the claim is true.

a.1 *more than two students* is increasing.
a.2 *fewer than two students* is decreasing.
b.1 *every student but John* is not increasing.
b.2 *every student but John* is not decreasing.
c.1 *either John or some student* is increasing.
c.2 *neither John nor any student* is decreasing.
d. *most but not all students* is neither increasing nor decreasing.
e. *exactly three students* is neither increasing nor decreasing.
f. *between five and ten students* is neither increasing nor decreasing.
g. *several students but no teachers* is neither increasing nor decreasing.

One observes that a conjunction or disjunction of an increasing DP with a decreasing one results in a non-monotonic DP (except where the result is trivial).

Proper Nouns. Proper nouns such as (Kim, P_0/P_1) were interpreted in Chapter 4 as the sort of Generalized Quantifier (GQ) called *individuals*. We repeat that definition here.

Definition 10.4. Given a domain E, for all $b \in E$ we define the GQ the *individual generated by* b (I_b) as: for all $p \in [E \to \{T, F\}]$, $I_b(p) = p(b)$. A generalized quantifier D is an *individual* iff for some $b \in E, D = I_b$.

Proper noun DPs, such as (Dana, P_0/P_1), not only satisfy the upward entailing paradigm in (15), their denotations, the individuals I_b, are clearly increasing. For suppose that $p \leq q$ and show that $I_b(p) = T$ implies that $I_b(q) = T$. But from the definition of I_b, if $I_b(p) = T$ then $p(b) = T$, whence $q(b) = T$, since $p \leq q$, so $I_b(q) = T$.

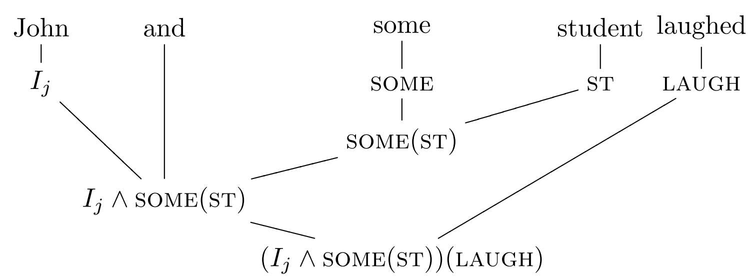
Exercise 10.5. Let I_E be the set of individuals over E. That is,

$$I_E = \{I_b \mid b \in E\}.$$

Define a bijection from I_E to E and conclude that $|I_E| = |E|$ (even if E is infinite).

Below we note some unexpected properties of (boolean compounds) of proper noun denotations so the reader should verify that he or she understands how to compute interpretations of boolean compounds involving proper nouns. Here is one illustration:

(18)



$$= I_j(\text{LAUGH}) \wedge \text{SOME(ST)}(\text{LAUGH})$$

Exercise 10.6. Exhibit, as in (18), interpretation trees for each of the following:

a. Either Mary or Sue laughed.
b. Neither John nor Mary cried.
c. Kim interviewed Sasha but not Adrian.

Exercise 10.7. Given $E = \{j, b, d\}$ we display the 8 properties over E, classifying them by the sets of objects they map to T. We exhibit the individuals I_j and I_b. You are to complete the table. Using the completed table say why $(I_j \wedge I_b)$ is not an individual.

p	$I_j(p)$	$I_b(p)$	$I_d(p)$	$(I_j \wedge I_b)(p)$	$(I_j \vee I_d)(p)$	$(I_j \wedge \neg I_d)(p)$
$\{j,b,d\}$	T	T				
$\{j,b\}$	T	T				
$\{j,d\}$	T	F				
$\{b,d\}$	F	T				
$\{j\}$	T	F				
$\{b\}$	F	T				
$\{d\}$	F	F				
$\emptyset$	F	F				

Observe that for $E = \{j,b,d\}$ there are just three individuals but $2^3 = 8$ properties functions from E into $\{T,F\}$. This is an instance of the first Counting Fact below. But first we observe that individuals, in distinction to other possible DP denotations, commute with the boolean functions. First, informally, we acknowledge the judgments in (19) of logical equivalence on English:

(19)
a. Boris doesn't smoke. ≡ It is not the case that Boris smokes.
b. Kim either laughed or cried. ≡ Either Kim laughed or Kim cried.
c. Sue both jogs and writes poetry. ≡ Sue jogs and Sue writes poetry.
d. Bob neither slept nor worked. ≡ Neither did Bob sleep nor did Bob work.

We have seen such judgments in earlier chapters, and we have also seen that various non-individual denoting DPs fail one or another of the paradigms above. For example, we lose logical equivalence if we replace *Sue* everywhere in (19c) by *some student*; we lose equivalence in (19b) replacing *Kim* with *every student*. A case we haven't looked at yet is negation, in (19a).

Exercise 10.8. Exhibit an informal model in which (a) and (b) below have different truth values:

a. More than two students didn't pass the exam.
b. It is not the case that more than two students passed the exam.

Once we think of proper names as DP functions mapping the set of functions $[E \to \{T,F\}]$ into $\{T,F\}$ the apparently trivial pattern in (19) becomes more interesting. Observe:

Theorem 10.2. *Individuals commute with all the boolean operations.*

Proof. Given a domain E, let b arbitrary in E. Then, for all properties $p, q \in [E \to \{T, F\}]$,

$$\begin{array}{llcll}
\text{a.} & I_b(\neg q) & = & (\neg q)(b) & \text{Def } I_b \\
 & & = & \neg(q(b)) & \text{Pointwise } \neg \\
 & & = & \neg(I_b(q)) & \text{Def } I_b \\
\text{b.} & I_b(p \vee q) & = & (p \vee q)(b) & \text{Def } I_b \\
 & & = & p(b) \vee q(b) & \text{Pointwise } \vee \text{ in } [E \to \{T, F\}] \\
 & & = & I_b(p) \vee I_b(q) & \text{Def } I_b \quad \square
\end{array}$$

Exercise 10.9.

a. On the pattern in part (b) above prove that individuals commute with $\wedge$.

b. Define the boolean function $\downarrow$ meaning *neither...nor...*, in terms of $\wedge$ and $\neg$.

c. State what it means for individuals to commute with *neither...nor...* then prove that they do.

Definition 10.5. For $\mathbf{A} = (A, \leq)$ and $\mathbf{B} = (B, \leq)$ boolean lattices, a function h from A into B is a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ iff h respects meets, joins, and complements, where

a. *h respects* $\wedge$ iff for all $x, y \in A$, $h(x \wedge y) = h(x) \wedge h(y)$.
When we write $x \wedge y$ above we are referring to the greatest lower bound of $\{x, y\}$ in A, since x, y are elements of A. And on the right, since $h(x)$ and $h(y)$ are elements of B, $h(x) \wedge h(y)$ is the glb of $f\{h(x), h(y)\}$ in B. A longer winded but sometimes helpful way of saying that h respects $\wedge$ is: if z is the glb of $\{x, y\}$ in A then $h(z)$f is the glb of $\{h(x), h(y)\}$ in B.

b. *h respects* $\vee$ iff for all $x, y \in A$, $h(x \vee y) = h(x) \vee h(y)$.

c. *h respects* $\neg$ iff for all $x \in A$, $h(\neg x) = \neg(h(x))$.

Terminology. A one to one (injective) homomorphism from $\mathbf{A}$ into $\mathbf{B}$ is called an *embedding* (also a *monomorphism*). A surjective (onto) homomorphism is called an *epimorphism* and a bijective homomorphism is called an *isomorphism*.

Exercise 10.10. For $\mathbf{A} = (A, \leq)$ and $\mathbf{B} = (B, \leq)$ boolean lattices and h a map from A to B,

a. Prove that if h respects $\wedge$ and $\neg$ then h respects $\vee$.

b. Prove the dual of (a): if h respects $\vee$ and $\neg$ then h respects $\vee$.

c. Prove: if h is a homomorphism then

   i. for all $x, y \in A$, $x \leq y \Rightarrow h(x) \leq h(y)$.
   (We often write $\Rightarrow$ for *if...then...*)

ii. $h(0_A) = 0_B$ and $h(1_A) = 1_B$.

Exercise 10.11. We define the binary *neither...nor...* function $\downarrow$ in any boolean lattice by $x \downarrow y =_{df} \neg x \wedge \neg y$.

Your task: For $\mathbf{A} = (A, \leq)$ and $\mathbf{B} = (B, \leq)$ boolean lattices, h a map from A into B,

a. Say what it means for h to respect $\downarrow$.
b. Prove: if h is a homomorphism then h respects $\downarrow$.
c. Prove: if h respects $\downarrow$ then h is a homomorphism (for this you must state $x \downarrow y$ in terms of $\neg$ and either $\wedge$ or $\vee$).

We see then that the individuals among the GQs are homomorphisms. Does the converse hold? That is, are all homomorphisms from P_1 denotations into P_0 denotations individuals? Surprisingly perhaps, and with one technical nuance, the answer is yes. First the nuance:

Definition 10.6. For $\mathbf{A}$ and $\mathbf{B}$ complete boolean lattices (so in each case every subset has a glb and every subset has a lub). Then a homomorphism h from A to B is said to be *complete* iff it respects all glbs and lubs. That is,

a. for all subsets K of A, $h(\bigwedge K) = \bigwedge\{h(k) \mid k \in K\}$, and
b. for all subsets K of A, $h(\bigvee K) = \bigvee\{h(k) \mid k \in K\}$. [3]

In fact each of conditions (a) and (b) implies the other, so to show that a homomorphism h is complete it suffices to show just one of them.

Theorem 10.3. *Let $\mathbf{B} = (B, \leq)$ be a boolean lattice. Then if every subset of B has a glb then every subset has a lub.*

Proof. Assume every subset of B has a glb. Then for K an arbitrary subset of B show that the lub of K is the glb of $\mathrm{UB}(K)$, the set of upper bounds for K. Dually, if all subsets have a lub, show that for any K, the glb of K is the lub of the set of lower bounds for K. We write this out more carefully in the Appendix to this chapter. $\square$

Exercise 10.12. State and prove the dual to Theorem 10.3.

Theorem 10.4. *A function h is a complete homomorphism*

$$h : [E \to \{T, F\}] \to \{T, F\}$$

iff $\exists b \in E$ such that $h = I_b$.

---

[3] The usual definition is slightly more general. It does not require the algebras to be complete and just says that whenever a subset K of A has a glb then $\{h(x) \mid x \in K\}$ has a glb in B and $h(\bigwedge K) = \bigwedge\{h(x) | x \in B\}$.

A proof of Theorem 10.4 is given in the Appendix.

Theorem 10.4 would allow us to define denotations for proper nouns of category P_0/P_1 without mentioning E. They would just be the complete homomorphisms from $\text{DEN}_E(P_1)$ into $\text{DEN}_E(P_0)$.

Counting Facts.

1. $|[A \to \{T, F\}]| = |\mathcal{P}(A)| = 2^{|A|}$
2. $|[A \to B]| = |B|^{|A|}$

Since we are going to be comparing the sizes of sets let us see why the first Counting Fact is true (A assumed finite, the case of interest). We have already shown, Chapter 8, that $|[A \to \{T, F\}]| = |\mathcal{P}(A)|$. This is so since each function g from A into $\{T, F\}$ determines a unique subset T_g of A, namely the set of elements in A which g maps to T. Different g correspond to different T_g, and each subset K of A is a T_g for some g. So the map sending each g to T_g is a bijection, so $|[A \to \{T, F\}]| = |\mathcal{P}(A)|$.

To see the idea behind the claim that $|\mathcal{P}(A)| = 2^{|A|}$ we give an induction argument. When $|A| = 0$, and so $A = \emptyset$, $\mathcal{P}(A) = \{\emptyset\}$ and so has cardinality $1 = 2^0$. Assume now that when $|A| = n$, $|\mathcal{P}(A)| = 2^n$. Then let us add one new element b to A forming a set $A \cup \{b\}$ of cardinality $n+1$. This set has double the number of subsets of A since each old subset K of A is a subset of $A \cup \{b\}$, and in addition for each old subset K we now have one new subset, $K \cup \{b\}$. So the number of subsets of $A \cup \{b\}$ is $2 \cdot 2^n = 2^{n+1}$. Thus for all $n \in \mathbb{N}$ if $|A| = n$ then $|\mathcal{P}(A)| = 2^n$, as was to be shown.

Observe that the number of individuals over a domain E of cardinality n is a vanishingly small number of GQ_E, the set of generalized quantifiers over E. Since the individuals over E correspond one for one with the elements of E, we have that $|E| = |\{I_b \mid b \in E\}|$. By Counting Fact 1 the number of properties of elements of E, $|[E \to \{T, F\}]|$ is $2^{|E|}$. And thus by Counting Fact 1 again the number of generalized quantifiers over E is $|[[E \to \{T, F\}] \to \{T, F\}]| = 2^{2^{|E|}}$

Thus in a model with just four entities, there are 4 individuals, $2^4 = 16$ properties and $2^{16} = 65,536$ GQs. Keenan and Stavi (1986) show that over a finite universe all GQs are actually denotable by English DPs. That is, for each GQ we can construct an (admittedly cumbersome) English DP which could denote that GQ. But the number of these GQs denotable by proper nouns is insignificant, just 4 out of 65,536 in the example above.

Counting Fact 2 is very useful and covers the first one as a special case. We are often interested in imposing linguistically motivated conditions on the elements of some $[A \to B]$ and want to evaluate how many of the functions in that set are ruled out by the conditions. Now, we see that Fact 2 holds when $|A| = 1$, since then there is one function for each element β of B (the one that maps the unique α in A to β). Suppose now the fact holds for all A of cardinality n, and we show it holds for any A with $|A| = n + 1$. So A is some $X \cup \{a\}$ for $|X| = n$ and $a \notin X$. By our assumption there are $|B|^n$ functions from X into B. And each of those functions corresponds to $|B|$ many new ones from A into B, according to the $|B|$ many ways each such f can take its value on the new α . Thus there are $|B| \cdot |B|^n = |B|^{n+1} = |B|^{|A|}$ functions from A into B, which is what we desired to show.

10.3 Semantic Generalizations

We turn now to some semantic generalizations we can make about English using our interpretative mechanisms as developed so far.

Generalization 1. Lexical DPs are always monotonic; almost always increasing, in fact almost always individual denoting.

Here is a snapshot of the lexical DPs of English: they include one productive subclass, the Proper Nouns: *John, Mary, ..., Siddartha, Chou en Lai, ...*. By "productive" we mean that new members may be added to the class without changing the language significantly. Lexical DPs also include listable sprinklings of (i) personal pronouns: *he/him,..* and their plurals *they/them*; (ii) demonstratives: *this/that* and *these/those*; and more marginally (iii) possessive pronouns: *mine/his/hers .../theirs*; and (iv) indefinite pronouns *everyone, everybody; someone/body*; and *no one, nobody*, though these latter appear syntactically complex. We might also include some DP uses of Dets, as *A good time was had by all*, *Some like it hot*, and *Many are called but few are chosen*, though we are inclined to interpret them as having an understood N *people* to account for their +human interpretation (a requirement not imposed by the Dets themselves).

Clearly the unequivocal cases of lexical DPs are increasing, mostly proper nouns. The only candidates for non-increasing lexical DPs are *few* and the "n" words (*no one*, *nobody*) and they are decreasing.

Query: Is there any reason why the simplest DPs should denote individuals? Is there any sense in which individuals are more basic that other DP denotations?

We make two suggestions towards an answer to the Query. First, there are many functions that a novel DP could denote—some 65,536 of them in a world of just 4 objects. But the denotations of syntactically complex DPs can be figured out in terms of the denotations of their parts. So the real learning problem for expressions concerns lexical ones, which have no parts and must be learned by brute force. Clearly learning is greatly simplified if the language learner can assume that lexical DPs denote individuals. In our model with 65,536 generalized quantifiers, only four (!) of them are individuals.

Second, the individuals do have a very special algebraic status among the GQs. Namely, every GQ can be expressed as a boolean function (ones defined in terms of meet, join and complement) of individuals! In other words, there is a sense in which all GQs are decomposable into individuals and the boolean operations. Moreover the boolean connectives are, as we have seen, polymorphic: for almost all categories C they combine expressions of category C to form further expressions in that category. So the denotation set of *most* C is a set with a boolean structure. This suggests that the boolean operations are not ones specific to one or another semantic category (one or another $\textsc{den}_{\mathcal{M}}(C)$) but rather represent, as Boole (1854) thought, properties of mind—ways we think about things rather than properties of things themselves.

This consideration is speculative. But the claim that all GQs are constructable by applying boolean operations to individuals is not. It is a purely mathematical claim whose proof we sketch here. In fact we have already seen, Exercise 10.7, that boolean functions of individuals may yield GQs that are not themselves individuals. All that is at issue is how many of these non-individuals are expressible in this way, and in fact all are. (To improve readability, slightly, we write 1 for T ("True") and 0 for F ("False"), as is standard).

Leading up to the proof, consider, for p a property, $\bigwedge\{I_b \mid p(b) = 1\}$. This GQ maps a property q to 1 iff every individual which is true of p is true of q, that is, iff $p \leq q$. (So this GQ is, in effect, "Every(p)".)

Now we want to build a boolean function of individuals which is, in effect, "No non-p". Taking the meet of that GQ with "Every(p)" will yield a GQ that holds just of p. Now "some non-p" is just $\bigvee\{I_b \mid (\neg p)(b) = 1\}$, so "no non-$p$" is the complement of that, namely $\neg\bigvee\{I_b \mid (\neg p)(b) = 1\}$, which is the same as $\bigwedge\{\neg I_b \mid (\neg p)(b) = 1\}$. (See Appendix). It holds of a property q iff for every b such that $(\neg p)(b) = 1$, $(\neg I_b(q)) = 1$, that is, $I_b(\neg q) = 1$, so $(\neg q)(b) = 1$. Thus it holds of q iff for every b, if $(\neg p)(b) = 1$ then $(\neg q)(b) = 1$, that is, iff $\neg p \leq \neg q$, which is equivalent to $q \leq p$. Combining these two results we have:

(20) $(\bigwedge\{I_b \mid p(b) = 1\} \wedge \bigwedge\{\neg I_b \mid (\neg p)(b) = 1\})(q) = T$
iff $\bigwedge(\{I_b \mid p(b) = 1\})(q) = 1$ and $\bigwedge(\{\neg I_b \mid (\neg p)(b) = 1\})(q) = 1$
iff $p \leq q$ and $q \leq p$
iff $p = q$.

For convenience call this GQ F_p. It is that GQ which holds just of p

and it is a boolean function of individuals. And one sees easily that for any GQ H,

(21) $H = \bigvee\{F_p \mid H(p) = T\}$.

Proof. Note that by the pointwise behavior of lubs, the join of a bunch of GQs maps a property q to 1 iff one of the GQs over which the join was taken maps q to 1. Now let q be an arbitrary property. We show that the functions on either side of the equation sign in (21) assign q the same value and are hence the same function. Suppose first that $H(q) = 1$. Then F_q is one of the F_p in the set over which the join is taken on the right, and since F_q maps q to 1 then the join does as well. So in this case the functions yield the same value. Suppose secondly that $H(q) = 0$. Then $F_q \notin \{F_p \mid H(p) = 1\}$, hence each of the F_p in that set maps q to 0, so their lub maps q to 0. So in this case as well H and $\bigvee\{F_p \mid H(p) = 1\}$ take the same value at q. As this covers all the cases our proof is complete. □

Thus any generalized quantifier is expressible as a boolean function of individuals. Of course (21) is not a definition of H. It is simply a truth about H, one that shows that an arbitrary GQ is identical to a boolean function of individuals [4].

Returning to Earth, Generalization 2 is a non-trivial empirical claim and also simplifies the learning problem for a language:

Generalization 2. Lexical Dets generally form monotonic DPs, almost always increasing.

Now many types of DPs are not monotonic:

(22) Some non-monotonic DPs: exactly five men, between five and ten students, about a hundred students, every/no student but John, every student but not every teacher, both John and Bill but neither Sam nor Mary, most of the students but less than half the teachers, either fewer than five students or else more than a hundred students, more boys than girls, exactly as many boys as girls.

Exercise 10.13. For each DP below exhibit an informal model which shows that it is not monotonic (not increasing and not decreasing)

a. exactly two students

[4] What we have shown is that the set of individuals over E is a set of complete generators for GQ_E. If E is finite it is provably a set of free generators for GQ_E. And actually Keenan and Faltz (1985) show more: any complete homomorphism from the set of individuals into any complete and atomic algebra extends to a complete homomorphism from GQ_E into that algebra.

b. John but not Mary

c. between five and ten students

d. every student but John

Thus the kinds of GQs denotable by the DPs in (20) are not available as denotations for DPs built from lexical Dets. So Generalization 2 is a strong semantic claim about natural language.

To support Generalization 2 observe that of the lexical Dets in (12) and (13), most clearly build increasing DPs: *each*, *every*, *all*, *some*, *my*, *the*, *this*, *these*, *several*, *a*, *both*, *many*, *most*. But *no*, *neither* and *few* build decreasing DPs. And opinions are less clear regarding bare numerals like *two*. In cases like *Are there two free seats in the front row?* we interpret *two free seats* as *at least two...*, which is increasing. In contexts like *Two students stopped by while you were out* the speaker seems to be using *two students* to designate two people he could identify, and as far as he himself (but not the addressee) is concerned he could refer to them as *they* or *those two students*. So this usage also seems increasing. Note that this usage is not paraphrasable by *Exactly two students stopped by while you were out*.

But in answer to *How many students came to the lecture?—Two*, the sense is "exactly two students", which is non-monotonic. We suggest here that the "exactly" part of this interpretation is due to the question, which in effect means "Identify the number of students that came to the lecture" rather than "Give me a lower bound on that number". So we favor Generalization 2 without the qualification "generally". Bare numerals are understood as increasing, and additional information provided by context can impose an upper bound on the number, forcing an "exactly", and thus a non-monotonic, interpretation.

But even if we take as basic the "exactly" interpretation of bare numerals it remains true that the GQs denotable by DPs of the form [Det + N], with Det lexical, are a proper subset of the set of denotable GQs. Reason: GQs denotable by DPs of the form *exactly n A's* are expressible as a conjunction of an increasing DP and a decreasing one: *Exactly n A's* denotes the same as *At least n A's and not more than n A's*, and Thysse (1983) has shown that the functions denotable by such DPs are just the convex [5] ones.

Definition 10.7. A GQ F is *convex* iff for all properties p, q, r, if $p \leq q \leq r$ and $F(p) = F(r) = T$ then $F(q) = T$.

So monotonic DPs are special cases of convex ones. But many DPs are not convex. Typical examples are disjunctions of increasing with

[5] Thysse calls them *continuous*, but we prefer *convex*.

decreasing DPs (*either fewer than six students or else more than ten students*) or disjunctions of properly convex ones (*either exactly two dogs or exactly four cats*). Also DPs like *more male than female students* are not convex. Thus in analogy with the distinction between lexical vs complex DPs we also see that there are functions denotable by complex Dets which are not denotable by lexical ones, examples being the functions denotable by *more male than female* and *either fewer than ten or else more than a hundred*.

The LFG: a case study in generalization The LFG, (17), is given with a pleasing level of generality. It makes sense to seek npi licensers in most categories, since, as it turns out, expressions in most categories are interpreted as maps from posets to posets. For example certain Prepositions license npi's, (23), and some Dets do within their N argument, (24).

(23) a. *He did it with *any* help.
b. He did it without *any* help.

(24) a. *Some student who has *ever* been to Pinsk really wants to return.
b. No student who has *ever* been to Pinsk really wants to return.
c. Every student who has *ever* been to Pinsk really wants to return.

But we do not pursue the empirical generalization here. Rather we are interested in the mathematical basis of the generalization itself. We sought, and found, a property that DP denotations shared with negation (whether S level or P_1 level). Note that their denotation sets share no elements. Negation denotes a map from a set to itself, and DPs denote maps from P_1 denotations to P_0 denotations. The common property was not some fixed element that they shared (such as deriving from Anglo-Saxon *ne*) but was the more abstract property of "being an order reversing (decreasing) function". The specific orders that *n't* and *at most two students* reverse are quite different.

10.4 Historical Background

Our treatment of negative polarity items draws on Ladusaw (1983), which in turn benefited from Fauconnier (1975). See also Zwarts (1981). Klima (1964) is the pioneering study of npi's. For more recent work see Nam (1994), Zwarts (1996), and Chierchia (2004). Our work on the boolean structure of natural language owes much to Keenan (1981) and Keenan and Faltz (1985). Montague (1974) was the first to treat DPs (our terminology) as generalized quantifiers (a term not used by him).

Recent overviews of work in generalized quantifier theory are Keenan (1996) and Keenan and Westerståhl (1996), Westerståhl (1985), Peters and Westerståhl (2006) and Keenan (2008). Useful collections of articles in this area are van der Does and van Eijck (1996), Kanazawa and Piñón (1994), Gärdenfors (1987), and van Benthem and ter Meulen (1985).

Appendix

Theorem 10.3. *Let* $\mathbf{B} = (B, \leq)$ *be a boolean lattice. Then if every subset of* B *has a glb then every subset has a lub.*

Proof. Let $K \subseteq B$ be arbitrary. Define the set $\mathrm{UB}(K)$ of upper bounds of K as follows.

$$\mathrm{UB}(K) = \{b \in B \mid b \text{ is an upper bound for } K\}.$$

So for $b \in \mathrm{UB}(K)$, for every $k \in K$, $k \leq b$. By the definition of glb then $k \leq \bigwedge \mathrm{UB}(K)$. $\mathrm{UB}(K)$ has a glb by by assumption that all subsets of B do. Therefore $\bigwedge \mathrm{UB}(K)$ is an upper bound for K. Now let b be an upper bound for K. So $b \in \mathrm{UB}(K)$. Trivially $\bigwedge \mathrm{UB}(K) \leq b$, so $\bigwedge \mathrm{UB}(K)$ is the least of the upper bounds for K, that is, $\bigwedge \mathrm{UB}(K) = \bigvee K$. □

Theorem 10.4. *The complete homomorphisms from* $[E \to \{T, F\}]$ *into* $\{T, F\}$ *are exactly the individuals* $\{I_b \mid b \in E\}$.

Proof.
($\Longleftarrow$) We have already seen that each I_b is a homomorphism. To show that such an h is complete we have: for any indexed family of P_1 denotations $\{p_j \mid j \in J\} \subseteq [E \to \{T, F\}]$,

$$\begin{array}{rcll} I_b(\bigvee_{j \in J} p_j) & = & (\bigvee_{j \in J} p_j)(b) & \text{Def } I_b \\ & = & \bigvee_{j \in J}(p_j(b)) & \text{Pointwise meets in } [E \to \{T, F\}] \\ & = & \bigvee_{j \in J}(I_b(p_j)) & \text{Def } I_b \end{array}$$

($\Longrightarrow$) Now let h be a complete homomorphism. We show that for some $b \in E$, $h = I_b$. Let p_b be that element of $[E \to \{T, F\}]$ that maps b to T and all other $b' \in E$ to F. h must map one of these p_b to T. If it maps all to F then it maps their lub to F. But $\bigvee_{b \in E} p_b$ is the unit element since it maps all $b \in E$ to T and as a hum, h must map the unit to the unit, which is T, a contradiction. So for some b, $h(p_b) = T$. And for q a map from E into $\{T, F\}$, $p_b \leq q$ iff $q(b) = T$. So h maps to T all the q that I_b maps to T. And if $q(b) = F$ then $h(q) = F$, otherwise $h(p_b \wedge q) = h(0) = T$. Thus $h = I_b$ since it holds of a q iff $q(b) = T$. □

The Infinite DeMorgan Laws. The proofs of (20) used an infinitary form of the DeMorgan laws. We prove these below, but first some lemmas.

Lemma 10.5. *For all boolean lattices B, all $x, y \in B$,*

$$x = x \wedge 1 = x \wedge (y \vee \neg y) = (x \wedge y) \vee (x \wedge \neg y).$$

Lemma 10.6 (The Finite De Morgan Laws)**.**

1. $\neg(x \vee y) = \neg x \wedge \neg y$, *and*
2. $\neg(x \wedge y) = \neg x \vee \neg y$.

Proof. We show (1) $\neg(x \vee y) = \neg x \wedge \neg y$.

$$\begin{array}{lrcl} \text{a.} & (x \vee y) \wedge (\neg x \wedge \neg y) & = & (x \wedge (\neg x \wedge \neg y)) \vee (y \wedge (\neg x \wedge \neg y)) \\ & & = & 0 \vee 0 \\ & & = & 0. \\ \text{b.} & (x \vee y) \vee (\neg x \wedge \neg y) & = & x \vee (y \vee (\neg x \wedge \neg y)) \\ & & = & x \vee ((y \vee \neg x) \wedge (y \vee \neg y)) \\ & & = & x \vee ((y \vee \neg x) \wedge 1) \\ & & = & x \vee \neg x \vee y \\ & & = & 1 \vee y \\ & & = & 1. \end{array}$$

Thus by Uniqueness of complements (Theorem 8.8),

$$(\neg x \wedge \neg y) = \neg(x \vee y). \qquad \square$$

We leave the dual lemma $\neg(x \vee y) = (\neg x \vee \neg y)$ as an exercise for the reader.

Note the following definition of a *dual*.

Definition 10.8. If φ is a statement in the language of boolean lattices then *dual(φ)* is the statement that results from simultaneously replacing all '$\wedge$' signs with '$\vee$' and all '$\vee$' with '$\wedge$', replacing all occurrences of '0' with '1' and '1' with '0', and all occurrences of '$\leq$' with '$\geq$' and all '$\geq$' with '$\leq$'.

Meta-Theorem: Duality. For φ a statement in the language of boolean lattices, $\varphi \equiv \text{dual}(\varphi)$.

The reason that Duality holds is that the dual of every axiom of boolean lattices is an axiom (though we didn't take $1 \neq 0$ as an axiom, but it obviously is logically equivalent to $0 \neq 1$). So whenever we have a proof of some ψ from the axioms then replacing each line in the proof by its dual is a proof of dual(ψ).

Lemma 10.7. *For all x, y in a boolean lattice B, $x \leq y$ iff $\neg y \leq \neg x$.*

Proof.

$x \leq y$	iff	$(x \wedge y) = x$	From def of glb
	iff	$\neg(x \wedge y) = \neg x$	Uniqueness of complements
	iff	$(\neg x \vee \neg y) = \neg x$	De Morgan Laws
	iff	$\neg y \leq \neg x$	From def of lub $\square$

Lemma 10.8 ((Weak) Infinite Distributivity)**.** *In any complete boolean lattice,*

$$x \wedge \bigvee_{j \in J} y_j = \bigvee_{j \in J} (x \wedge y_j).$$

Proof. To enhance readability write when convenient y for $\bigvee_{j \in J} y_j$.

($\Longleftarrow$) For each j, $y_j \leq y$, so $x \wedge y_j \leq x \wedge y$, so $x \wedge y$ is an ub for $\{x \wedge y_j \mid j \in J\}$ thus

$$\bigvee_{j \in J} (x \wedge y_j) \leq x \wedge y = x \wedge \bigvee_{j \in J} y_j.$$

($\Longrightarrow$) We show that $x \wedge y \leq$ every upper bound for $\{x \wedge y_j \mid j \in J\}$. Let u be such an ub. Then for each j, $y_j = (x \wedge y_j) \vee (\neg x \wedge y_j) \leq u \vee \neg x$. This is because $(x \wedge y_j) \leq u$ and $(\neg x \wedge y_j) \leq \neg x$. Thus $u \vee \neg x$ is an ub for $\{y_j \mid j \in J\}$ so $y \leq u \vee \neg x$, whence $x \wedge y \leq x \wedge (u \vee \neg x) = x \wedge u \leq u$. Since u was an arbitrary ub for $\{x \wedge y_j \mid j \in J\}$ we have that

$$x \wedge y \leq \bigvee_{j \in J} (x \wedge y_j),$$

proving equality. $\square$

There are stronger forms of distributivity but they do not hold in all complete boolean lattices.

Theorem 10.9 (The Infinitary DeMorgan Laws.)**.**

1. $\neg \bigvee_{k \in K} k = \bigwedge_{k \in K} \neg k$.
2. $\neg \bigwedge_{k \in K} k = \bigvee_{k \in K} \neg k$.

Proof.

1. ($\Longrightarrow$)For each $k \in K$, $k \leq \bigvee_{k \in K} k$, so $\neg \bigvee_{k \in K} k \leq \neg k$, so $\neg \bigvee_{k \in K} k$ is a lb for $\{\neg k \mid k \in K\}$, so

$$\neg \bigvee_{k \in K} k \leq \bigwedge_{k \in K} \neg k.$$

($\Longleftarrow$) Write d for $\bigwedge_{k \in K} \neg k$. We are to show that $d \leq \neg \bigvee_{k \in K} k$. Observe that for each $k \in K$, $k \wedge d = 0$, since $k \wedge d \leq k \wedge \neg k = 0$.

Thus (using the general boolean fact that $x \leq y$ iff $x \wedge \neg y = 0$) $k \leq \neg d$. Since k was arbitrary in K, $\bigvee_{k \in K} k \leq \neg d$, whence

$$d = \bigwedge_{k \in K} \neg k \leq \neg \bigvee_{k \in K} k,$$

as was to be shown.

2. The proof of the dual statement is left to the reader. $\square$

11

Semantics V: Classifying Quantifiers

In this chapter we present several semantically defined subclasses of *declarative count* Determiners in English and establish a variety of empirical and mathematical generalizations about them. Some of these concern the semantic characterization of syntactic phenomena, others are purely semantic. At various points we note properties they share with *interrogative quantifiers* (1b), *mass quantifiers* (1c) and *adverbial quantifiers* (1d), classes that are less well understood than the Dets (1a) we focus on.

(1)
a. *All* / *Most* / *No* poets daydream.
b. *How many* / *Which* / *Whose* children are laughing?
c. There was *(too) much* / *(not) enough* salt in the soup.
d. I *always* / *usually* / *often* / *rarely* / *never* work on weekends.

Useful sources of information about the expression of quantification in diverse languages are Bach et al. (1995), Gil (1993), and Matthewson (2008).

A notational preliminary. Following the lead of most work in Generalized Quantifier Theory we treat P_1s simply as subsets of a domain E rather than functions from E into $\{T, F\}$. Up to isomorphism they are the same thing, recall. And more generally n-ary relations on E are treated as subsets of E^n . Further we refer to DPs (*Mary*, *Most poets*,...) as denoting functions of type (1), called generalized quantifiers, rather than using $((\mathsf{e}, \mathsf{t}), \mathsf{t})$ which gets cumbersome when quantifiers of higher type are studied (as here). A quantifier of type (1) is a function from subsets of E^1 into $\{T, F\}$. Dets like *all*, *most*, etc. are of type (1,1), mapping subsets of E to functions of type (1). Quantifiers of type (2) map binary relations—subsets of E^2—into $\{T, F\}$. And quantifiers of type ((1,1),1) map pairs of subsets of E to functions of type (1). This type notation derives from Lindström (1966). The interpretation of (2a) below is now given as (2b):

(2) a. Most poets daydream.
b. (MOST(POET))(DAYDREAM).

Here both POET and DAYDREAM are subsets of the domain. (We see shortly though that the role of the noun property, POET, in the interpretation (2a) is quite different from that of the predicate property DAYDREAM). POET in these cases is sometimes called the *restriction* (of the quantifier). Using this set notation[1] here are some further definitions of English quantifiers of type (1,1). We have already seen most of them. Read the first line in (3) as "some A's are B's iff the intersection of the set of A's with the set of B's is not empty". The other lines are read analogously.

(3)	a.	$\text{SOME}(A)(B) = T$	iff $A \cap B \neq \emptyset$
	b.	$\text{NO}(A)(B) = T$	iff $A \cap B = \emptyset$
	c.	$(\text{EXACTLY TWO})(A)(B) = T$	iff $\lvert A \cap B\rvert = 2$
	d.	$(\text{ALL BUT ONE})(A)(B) = T$	iff $\lvert A - B\rvert = 1$
	e.	$\text{MOST}(A)(B) = T$	iff $\lvert A \cap B\rvert > \lvert A\rvert/2$
	f.	$(\text{THE } n)(A)(B) = T$	iff $\lvert A\rvert = n$ and $A \subseteq B$

We turn now to the semantic classification of English Dets. We offer a field guide to English Dets in an appendix to this chapter. Those Dets serve to indicate the syntactic and semantic diversity of the expressions of interest here.

[1] The "set" approach we take does force us to ignore the value judgment Dets such as *too many*, *not enough*, *surprisingly few*, etc. in the Appendix. Such Dets are inherently intensional and so cannot be treated a functions whose domain is simply the sets of objects with given properties.

For example in a model in which it happens that the doctors and the lawyers are the same individuals Ss (a) and (b) below can still have different truth values:

a. Not enough doctors attended the meeting.
b. Not enough lawyers attended the meeting.

Imagine for example that we a discussing a meeting of the American Medical Association. 500 doctors are required for a quorum, but only one lawyer is required, to take the minutes. And suppose that just 400 doctor-lawyers shows up. (a) is true and (b) is false. Note that if *not enough* is replaced we by *every* (making the nouns singular) the resulting Ss must have the same truth value. This says that *every* is extensional, meaning that it can be treated as a function whose arguments are the extensions of the property denoting expressions, that is, the sets of individuals which have the property.

11.1 Quantifier Types

The most widely studied quantifiers in English are those of type (1,1). We distinguish four subcases, illustrated in (4a,b,c,d), according to what we need to know to evaluate the truth of the sentences they build:

(4) a. *Some* students read the Times.
b. *All* students read the Times.
c. *Most* American teenagers read the Times.
d. *The four* students read the Times.

To determine the truth of (4a) it suffices to have at hand the intersection of the set of students with the set of people who read the Times. If that set is empty (4a) is false, otherwise it is true. To establish the truth of (4a) we need know nothing about students who didn't read the Times, nor anything about people who read the Times who are not students. But knowing which students read the Times does not decide the truth of (4b). For that we must know about the set of students who don't read the Times. If that set is non-empty then (4b) is false, otherwise it is true. And to evaluate the truth of (4c) we must know both about the students who read the Times and also about the students who don't, to verify that the former outnumber the latter. Lastly, (4d) requires that we know the cardinality of the restrictor STUDENT and that the set of students who don't read the Times is empty. Note that merely knowing the cardinality of the set of students who read the Times does not suffice. These observations determine the *intersective* (or *generalized existential*), *co-intersective* (or *generalized universal*), *proportional* and *definite* classes of Dets respectively.

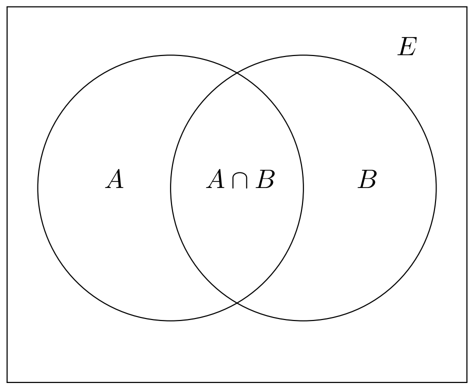
11.1.1 Intersective Dets

Definition 11.1. *Intersective* (or *generalized existential*) Dets are ones whose denotations D satisfy:

$$D(A)(B) = D(X)(Y) \text{ whenever } A \cap B = X \cap Y.$$

This invariance condition is a way of saying that the value D assigns to A, B just depends on $A \cap B$. This value is unchanged (invariant) under replacement of A by X and B by Y provided X has the same intersection with Y as A does with B. It is easy to see that SOME is intersective. Given $A \cap B = X \cap Y$ then either both intersections are empty and $\text{SOME}(A)(B) = \text{SOME}(X)(Y) = F$, or both are non-empty and $\text{SOME}(A)(B) = \text{SOME}(X)(Y) = T$.

But Definition 11.1 is more general than meets the eye since it does not make any commitment as to precisely what type of value D assigns to a pair A, B. Mostly here we think of it as a truth value, but on richer semantic theories in which Ss denote *propositions*, functions from possible worlds to truth values, $D(A)(B)$ would denote a proposition. And for some of our examples below we see that $D(A)(B)$ is whatever it is that questions denote. A Venn diagram might be helpful in understanding Definition (11.1).



So to say that a Det is intersective is to say that when it looks at a pair of sets A, B it makes its decision just by considering the area where they overlap, $A \cap B$. Theorems 11.1a,b are often helpful in establishing the intersectivity of a Det. (E as always is the understood domain).

Theorem 11.1.

a. *D is intersective iff for all sets A, B $DAB = D(A \cap B)(E)$. We may read "$D(A \cap B)(E)$" as "D As that are Bs exist".*
b. *D is intersective iff for all A, B $DAB = D(E)(A \cap B)$. We may read "$D(E)(A \cap B)$" as "D individuals are both As and Bs".*

Proof.

a. ($\Longrightarrow$) Given D intersective the right hand side of the "iff" above holds since $(A \cap B) \cap E = A \cap B$, since $A \cap B$ is a subset of E. ($\Longleftarrow$) Going the other way, assume the right hand side and suppose that $A \cap B = X \cap Y$. But then $D(A)(B) = D(A \cap B)(E)$ by the assumption, which equals $D(X \cap Y)(E)$ since $A \cap B = X \cap Y$, and equals $D(X)(Y)$ again by the assumption, so D is intersective. □

Exercise 11.1. Prove Theorem 11.1b.

The Dets in (5) are intersective since the (x,x') pairs are logically equivalent.

(5) a. Some students are vegetarians.
a′. Some students who are vegetarians exist.
b. Exactly ten boys are on the team.
b′. Exactly ten individuals are boys on the team.
c. How many athletes smoke?
c′. How many individuals are athletes who smoke?
d. Which students sleep in class?
d′. Which individuals are students who sleep in class?

Intersective Dets are the most widely distributed of the English Dets. Here are some examples, in which we include the interrogative Dets *Which?* and *How many?*, though most intersective Dets are declarative.

(6) Some intersective Dets: *some, a/an, no, several, more than six, at least six, exactly six, fewer than six, at most six, between six and ten, just finitely many, infinitely many, about a hundred, a couple of dozen, practically no, nearly twenty, approximately twenty, not more than ten, at least two and not more than ten, either fewer than five or else more than twenty, that many, How many?, Which?, more male than female, just as many male as female, no...but John*

The intersective Dets present some interesting semantic subcategories, but first let us see that there are many Dets that are not intersective. *Every* for example is not:

(7) Let $A = \{2, 3\}$, $B = \{1, 2, 3, 4\}$ and let $A' = \{1, 2, 3\}$ and $B' = \{2, 3, 4\}$. Then clearly $A \cap B = \{2, 3\} = A' \cap B'$. But since $A \subseteq B$ we have that $\text{EVERY}(A)(B) = T$. But $A' \nsubseteq B'$ so $\text{EVERY}(A')(B') \neq T$. Thus EVERY is not intersective.

Exercise 11.2. For each Det below exhibit an informal model as in (7) which shows that it is not intersective.

a. all but two b. most c. the two

The productivity of the intersectivity class of Dets prompts us to wonder just how many of the possible Det denotations—maps from $\mathcal{P}(E)$ into $[\mathcal{P}(E) \to \{T, F\}]$—are intersective. That is, just how strong is the intersectivity condition? The answer is somewhat surprising. Here are two observations leading up the relevant result.

First, recall that the set of possible Det denotations is a boolean lattice with the operations given pointwise. This predicts, correctly, the following logical equivalences:

(8) (Some but not all) cats are black.
$\equiv$ (Some cats but (not all) cats) are black.

$\equiv$ (Some cats but (not (all cats))) are black.
$\equiv$ Some cats are black but (not (all cats)) are black.
$\equiv$ Some cats are black but not (all cats are black).

And it turns out that the set of intersective functions in this class is closed under these boolean operations. That is, if D and D' are intersective, so are $(D \wedge D')$, $(D \vee D')$, and $\neg D$. So *at least two and not more than ten* is intersective since *at least two* and *more than ten* are. Generalizing, the set INT_E of intersective functions over E is a boolean lattice. This result appears a little "abstract" but in fact it is easy to prove (though for reasons of space in this chapter we will not be concerned to prove such results). Let us see for example, as claimed above, that the join of two intersective functions is itself intersective:

Fact 1. Given E, let D and D' be intersective Dets. We show that $(D \vee D')$ is intersective.

Proof. Let $A \cap B = X \cap Y$. We must show that

$$(D \vee D')(A)(B) = (D \vee D')(X)(Y).$$

We calculate:

$$\begin{array}{lcll} (D \vee D')(A)(B) & = & (D(A) \vee D'(A))(B) & \text{Pointwise } \vee \text{ in Det} \\ & = & (D(A)(B) \vee D'(A)(B)) & \text{Pointwise } \vee \text{ in } \mathrm{GQ}_E \\ & = & (D(X)(Y) \vee D'(X)(Y)) & D, D' \text{ are intersective} \\ & = & (D(X) \vee D'(X))(Y) & \text{Pointwise } \vee \text{ in } \mathrm{GQ}_E \\ & = & (D \vee D')(X)(Y) & \text{Pointwise } \vee \text{ in Det} \end{array}$$

The first and last lines establish that $(D \vee D')$ is intersective. $\square$

Exercise 11.3. Using a "pointwise" argument as in Fact 1, show that INT_E is closed under (a) meets, and (b) complements.

Thus we see that INT_E is a boolean lattice (a sublattice of the lattice of Det functions of type (1,1)). And we have the somewhat surprising result Keenan (1993):

Theorem 11.2. *For any domain E, INT_E, the set of intersective functions of type (1,1), is isomorphic to GQ_E, the set of functions of type (1). The map sending each intersective D to $D(E)$ is the desired isomorphism*

So, semantically, the isomorphic image of *some* is *some individual*, of *less than ten*, *less than ten individuals*, etc. This result is surprising, as possible Det denotations map $\mathcal{P}(E)$ into GQ_E, and so in general vastly outnumber GQ_E. But when we limit ourselves to intersective Det denotations we see that, up to isomorphism, they are just the familiar generalized quantifiers. Recalling that isomorphic structures are elementarily equivalent, that is, they make the same sentences true, we can say:

(9) The intersective Dets and the generalized quantifiers over an E have the same logical expressive power.

So if the only Dets in English were intersective there would be little logical point in distinguishing between them and generalized quantifiers. But as we have seen, English presents many non-intersective Dets.

Finally, using the Counting Facts of the previous chapter we can see what portion of the logically possible Det denotations (functions of type (1,1)) are intersective:

Fact 2. Given $|E| = n$, the number of logically possible (declarative) Det denotations is

$$|[\mathcal{P}(E) \rightarrow [\mathcal{P}(E) \rightarrow \{T, F\}]]| = 2^{4^n}.$$

In contrast,

$$|\mathrm{GQ}_E| = |[\mathcal{P}(E) \rightarrow \{T, F\}]| = 2^{2^n}.$$

So in a model with just two individuals, $|E| = 2$, there are $2^4 = 16$ distinct generalized quantifiers and so by Theorem 11.1 just $2^4 = 16$ intersective functions. But there are $2^{16} = 65,536$ logically possible (declarative) Det denotations.

Now observe that most of the Dets in (6) satisfy a condition stronger than intersectivity, they are *cardinal*: their value depends not on which objects lie in the intersection of their arguments but merely on *how many* objects lie in that intersection.

Definition 11.2. A possible Det denotation D is *cardinal* iff for all properties A, B, X, Y

$$DAB = DXY \text{ if } |A \cap B| = |X \cap Y|.$$

Dets such as *some*, *no*, *practically no*, *more than / less than / exactly ten*, *between ten and twenty*, *just finitely many*, *infinitely many*, *about a hundred*, *How many?* are cardinal. SOME is cardinal. (A definition equivalent to (3) is $\text{SOME}(A)(B) = T$ iff $|A \cap B| > 0$). The intersective Dets include "vague" ones, such as *about twenty* and *practically no*, since, while we may be uncertain concerning precisely how many sparrows must be on your clothesline in order for there to be about twenty, it does seem clear that

(10) If the number of sparrows on my clothesline is the same as the number of students in my yoga class then *About 20 sparrows are on my clothesline* and *About 20 students are in my yoga class* must have the same truth value.

Further, boolean compounds of cardinal Dets are themselves cardinal. So at *least two and not more than ten* is cardinal since *at least two* and *more than ten* are. And CARD_E, the set of cardinal (declarative) Dets over E, is a very small subset of INT_E.[2]

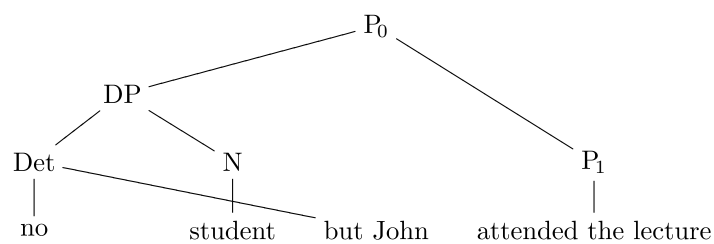
Theorem 11.3. *For* $|E| = n$, $|CARD_E| = 2^{n+1}$.

For example given an E with just 3 elements, there are $2^8 = 256$ intersective Dets, but only $2^4 = 16$ cardinal ones. Indeed, almost all the Dets in (6) are cardinal, so it is reasonable to ask whether English has any Dets that are intersective but not cardinal. Our best examples are those in (11), where only (11a) is syntactically simple:

(11) a. *Which* students attended the lecture?
b. *No* student but John came to the lecture.
c. *More male than female* students came to the lecture.

Clearly (11a) asks us to identify the students who attended, not merely to say how many there were. So if we just know how many students attended the lecture we do not have enough information to answer (11a) or to decide the truth of (11b) or (11c). (11b) is not unreasonably represented by the discontinuous Det *no...but John*, as in (12):

(12)



And we may interpret *no...but John* as in:

(13) $(\text{NO...BUT JOHN})(A)(B) = T$ iff $A \cap B = \{John\}$.

That is, "No A but John is a B" says that the As who are Bs consist just of John. So the value of this Det at a pair A, B is decided by its intersection, but it has to see that the intersection is $\{John\}$, not for example $\{Mary\}$, and not merely how many objects are in the intersection. So *no...but John* is intersective but not cardinal. Similarly *more male than female* is not cardinal, since in (11c) its value depends not simply on how many students came to the lecture, but rather on how many male students came and how many female students came.

[2] Assume that E is finite with cardinality n. Then there are $n + 1$ functions of the form (EXACTLY k) where $0 \leq k \leq n$. Each cardinal Det = a join ("disjunction") of functions of the form (EXACTLY k), so there are as many of them as there are subsets of this set of $n+1$ functions. That is, there are 2^{n+1} cardinal Det functions.

We Turn now to some classes of non-intersective Dets.

11.1.2 Co-Intersective Dets

Co-intersective (generalized universal) Dets depend on $A - B$, the As that are not Bs, just as intersective Dets depend on $A \cap B$. They are defined formally below, and exemplified in (14).

Definition 11.3. A Det D is *co-intersective* iff

$$DAB = DXY \text{ whenever } A - B = X - Y.$$

(14)	a.	$\text{ALL}(A)(B) = T$	iff $A - B = \emptyset$
	b.	$\text{ALL-BUT-SIX}(A)(B) = T$	iff $\lvert A - B\rvert = 6$
	c.	$\text{ALL-BUT-FINITELY-MANY}(A)(B) = T$	iff $A - B$ is finite
	d.	$\text{EVERY...BUT-JOHN}(A)(B) = T$	iff $A - B = \{j\}$

Note that for any sets A, B $A \subseteq B$ iff $A - B = \emptyset$, so the definition of ALL above is as previously defined though the definitions are stated differently. And we may define the *co-cardinal* Dets as those D satisfying $DAB = DXY$ whenever $|A - B| = |X - Y|$. The Dets in (14a,b,c) are co-cardinal, that in (14d) is not.

Theorem 11.4. *The set CO-INT$_E$ and CO-CARD$_E$ of co-intersective and co-cardinal functions over E are both closed under the pointwise boolean operations and thus form boolean lattices. They have the same size as INT$_E$ and CARD$_E$ respectively.*

Thus *not all* and *all but two or all but four* are co-intersective since the items negated and disjoined are.

The value of a co-intersective Det at properties A, B is decided by a single property, $A - B$. So again the map sending each co-intersective D to $D(E)$ is an isomorphism from the co-intersective functions to the set of generalized quantifiers, whence the intersective and the co-intersective Dets are isomorphic) . But even taken together INT$_E$ and CO-INT$_E$ constitute a minuscule proportion of the possible Det denotations: e.g. for $|E| = 2$ just 30 of the 65,536 functions from $\mathcal{P}(E)$ into GQ$_E$ are either intersective or co-intersective. We note that the two trivial Dets, **0** and **1**, which map all A, B to F and all A, B to T respectively are the only Det denotations which are both intersective and co-intersective.

The co-intersective Dets are structurally less diverse than the intersective ones. They are basically just the universal quantifier with exception phrases, including *almost all.* Also we seem to find no clear examples of interrogative Dets which are co-intersective. Our best example is *All but how many?* as in (15a), of dubious grammaticality. It means the same as the natural (15b) built from the intersective *How* *many?*.

(15) a. ??All but how many students came to the party?
b. How many students didn't come to the party?

11.1.3 Proportionality Dets

A third natural class of Dets that take us well outside the generalized existential and universal ones are the proportionality Dets.

Proportionality Dets decide DAB according to the proportion of As that are Bs. So, they yield the same value at A, B as they do at X, Y when the proportion of As that are Bs is the same as the proportion of X's that are Y's. Some examples first, then the definition, then some illustrative denotations:

(16) a. *Most* poets daydream.
b. *Seven out of ten* sailors smoke Players.
c. *Less than half* the students here got an A on the exam.
d. *At most ten per cent of* the students passed the exam.
e. *Between a third and two thirds* of the students are vegetarians.
f. *All but a tenth of* the students are vegetarians.
g. *Not one student in ten* can answer that question.

Definition 11.4. A Det D is *proportional* iff

$$DAB = DXY \text{ whenever } |A \cap B|/|A| = |X \cap Y|/|X|.$$

(17) a. $\textsc{seven-out-of-ten}(A)(B) = T$ iff $10 \cdot |A \cap B| = 7 \cdot |A|$
b. $\textsc{at-most-}10\%(A)(B) = T$ iff $10 \cdot |A \cap B| \leq |A|$
c. $\textsc{all-but-a-tenth}(A)(B) = T$ iff $10 \cdot |A - B| = |A|$
d. $\textsc{not-one...in-ten}(A)(B) = T$ iff $10 \cdot |A \cap B| < |A|$

Co-proportional Dets such as *all but a tenth* are covered by Definition 11.4 since the proportion of As with B is uniquely determined by the proportion that lack B: *All but (at most) a tenth* means the same as *(At least) nine out of ten*, so (18a,b) are logically equivalent.

(18) a. All but at most a tenth of the students read the Times.
b. At least nine out of ten students don't read the Times.

Fractional and percentage Dets, ones of the form n *out of* m, $n \leq m$, and the discontinuous *less than/more than/just one...in ten* are proportional. They have been little studied. Our best candidates for interrogative proportional Dets are *What percentage / proportion of*?

Proportionality Dets are "typically" neither intersective nor co-intersective, but there are a few exceptions: *exactly zero per cent* means *no* and *more than zero per cent* means *at least one*, so both are intersective; *a hundred per cent* means *all* and *less than a hundred per cent* means *not all*, so both are co-intersective. So the proportionality Dets have a small non-empty intersection with the (co)-intersective Dets.

The three classes of Dets so far discerned are "booleanly natural" in that boolean compounds of Dets in the class yield Dets in the class. But the classes are non-exhaustive: there are Dets in English which fall in none of these classes. The definite Dets defined below (based on Barwise and Cooper (1981)) in general lie outside the classes so far discerned. They are also not booleanly natural. Some examples[3] first:

(19)
a. $\text{THE-TWO}(A)(B) = T$ iff $|A| = 2$ and $\text{ALL}(A)(B) = T$
b. $\text{THE-TWO-OR-MORE}(A)(B) = T$ iff $|A| \geq 2$ and $\text{ALL}(A)(B) = T$
c. $\text{JOHN'S-TWO}(A)(B) = T$ iff $|A \text{ WHICH JOHN HAS}| = 2$ and $\text{ALL}(A \text{ WHICH JOHN HAS})(B) = T$

(Note that A WHICH JOHN HAS is a subset of A).

Now we define, tediously:

Definition 11.5.

a. A possible Det denotation D is *definite* iff for all $A \subseteq E$, either $DAB = F$ all B, or $DA = \text{ALL}(C)$, some non-empty $C \subseteq A$.
   D is called *definite plural* if the C in this last condition must always have at least two elements.
b. A Det d is *definite (plural)* iff there are models $\mathcal{M}$ in which $[\![d]\!]_{\mathcal{M}}$ is non-trivial and for all such $\mathcal{M}$, $[\![d]\!]_{\mathcal{M}}$ is definite (plural).

Note that *all* is not definite. It is non-trivial but it fails the disjunction "$DAB = F$ all B, or $DA = \text{ALL}(C)$, some non-empty $C \subseteq A$". It fails the first disjunct since for every A, $\text{ALL}(A)(A) = T$. It fails the second since for $A = \emptyset$ there is no non-empty subset C of A.

In contrast the reader can compute that *the five* and *John's five* are definite plural.

A syntactic query: Which DPs occur naturally in the post *of* position of partitives of the form: *two of* ___?

[3]Other Dets such as *these* and *your* can be subsumed under our definition of *definite plural*, but their deictic character (the interpretative dependency on the context of utterance) would introduce a very non-trivial dimension to our definition of interpretation in a model, one that we lack the space, and knowledge, to present here.

Note that (THE TWO) = (EXACTLY TWO) $\wedge$ ALL; (THE TWO OR MORE) = (AT LEAST TWO) $\wedge$ ALL. So extensionally these definite Dets are meets of intersective with co-intersective ones. But they differ in that *the two cats* presupposes that there exist two cats rather than merely asserting as, as given in (EXACTLY TWO) $\wedge$ ALL.

For example, *two of the ten students* and *two of John's five cats* are natural, whereas **two of most students*, **two of no students* certainly are not. Also this position seems closed under conjunctions and disjunctions but not under negation or *neither…nor…*: *two of John's poems or (of) Mary's plays were accepted*; *several of the students and (of) the teachers attended*. But **two of not the ten students* and **two of neither the ten students nor the ten teachers*. The same pattern of acceptability obtains replacing *two* by other plural, count Dets like *several*, *more than ten*, *all but ten*, *a majority of*, etc. So we propose to answer the query above with (20).

(20) The DPs occurring in *two of* ___ are just those in the closure under conjunction and disjunction of the DPs of the form Det + N, with Det definite plural as above.

We defined our frame with *two of* ___ because some Dets allow a mass interpretation with a singular terms in the post-of position (but are understood as count terms with plural DPs. Compare *All / Most of the house was unusable after the flood* versus *All / Most of the houses were unusable after the flood.* See Higginbotham (1994) and references cited there for a conceptually pleasing treatment of mass terms.

11.1.4 "Logical" Dets

Dets such as *John's ten*, *no…but John*, and *more male than female* lack the "logical" character of classical quantifiers and some might hesitate to call them quantifiers. But terminology aside, the challenging question here is whether we can characterize what is meant by a "logical" character. We can. This character is captured by the idea that the interpretations of "logical" expressions (not just Dets) remain invariant under structure preserving maps of the semantic primitives of a model. Such maps are the isomorphisms from a structure to itself, called *automorphisms.* For example an automorphism of a boolean lattice $(B, \leq)$ is a bijection h of B which fixes the $\leq$ relation: $x \leq y$ iff $h(x) \leq h(y)$). The primitives of a standard model $\mathcal{M}$ are just its domain $E_{\mathcal{M}}$ and the boolean lattice of truth values $\{T, F\}$. The only automorphism of $\{T, F\}$ is the identity map, as it is the only bijection on $\{T, F\}$ which fixes the implication order. The automorphisms of E are all the permutations (bijections) of E as E is not endowed with any relations or functions, so any permutation of E preserves "all" its structure. Thus an automorphism of $(E, \{T, F\})$ is in effect just a permutation of E (we omit the subscript $\mathcal{M}$). Moreover each such automorphism h lifts to an automorphism of the denotation sets built from E and $\{T, F\}$ in a standard way: an automorphism h maps each subset K of E to $\{h(k) \mid k \in K\}$; it maps an ordered pair (x, y) of elements of E to $(h(x), h(y))$, and it maps a binary relation R on E to $\{h(x, y) \mid (x, y) \in R\}$, etc.

An object in any denotation set is called *automorphism invariant* if it is mapped to itself by all the automorphisms. These are the objects you cannot change without changing structure. They are the ones with a "logical" character. For example the only subsets of E which are automorphism invariant are E and $\emptyset$, the denotations for *exist* and *not exist*. The invariant binary relations are $\emptyset$, $E \times E$, $\{(x, x) \mid x \in E\}$ and $\{(x, y) \in E^2 \mid x \neq y\}$, these latter two being the denotation of *is* (or *equals*) and *not equals*, respectively. Proceeding in this way (Keenan and Westerståhl (1996)) we see that

(21) The denotations of Dets such as *every*, *some*, *no*, *the two*, *most*, *all but five*, *two thirds of the*, etc. are automorphism invariant, those of *John' five*, *every...but John*, *more male than female* may denote functions which are not automorphism invariant.

We'll see shortly that automorphism invariance improves our understanding of the relation between intersective and cardinal Dets, as well as that between co-intersective and co-cardinal ones. For later reference we note:

Theorem 11.5. *The set of type (1,1) functions,*

$$[\mathcal{P}(E) \to [\mathcal{P}(E) \to \{T, F\}]],$$

is a boolean lattice pointwise. The subset of automorphism invariant (AI) functions is a sublattice of it. That is, the AI functions are closed under $\wedge$, $\vee$, *and* $\neg$.

11.2 Generalizations Concerning Det denotations

Surveying the Dets considered so far we see that a few, namely the intersective ones, make their truth decision at a pair A, B of sets just by checking $A \cap B$; a few others just check $A - B$; yet others, the proportionality Dets, check both $A \cap B$ and $A - B$, and finally the definite Dets consider both A and $A - B$, and partitive Dets check A and then otherwise behave like intersective, universal, or proportional Dets, as in: *two / all / most of the ten*. We can generalize from these observations to the claim that in English, evaluating the interpretation of Det As are Bs does not require that we look outside the As. This observation is formalized in the literature by two independent conditions: *Domain Independence* and *Conservativity*, both of which are non-trivial semantic properties of Det denotations.

11.2.1 Domain Independence

Domain Independence (DI) is a global condition on Dets, comparing their denotations in different universes. It was first proposed as a semantic universal of natural language in van Benthem (1984a)[4]. It is prominent in Data Base Theory (see Abiteboul et al. (1995) pp. 77) and rules out putative Det denotations like BLIK below, which we think of as associating with each domain E a type (1,1) function BLIK_E on E:

(22) For all E, $\text{BLIK}_E(A)(B) = T$ iff $|E - A| = 3$.

Were Blik an English Det then *Blik cats can fly* would be true iff the number of non-cats was three. And we could change the truth value of the S merely by adding some non-cats to the domain, which is unnatural. Note though that (22) assumes that with each E, BLIK associates a single quantifier. But, anticipating slightly, some Dets, like *John's two*, can have several denotations over a given domain varying with the individual *John* denotes and the things he "has". So we define:

Definition 11.6 (Domain Independence (DI))**.** A functional D associating each domain E with a family of functions of type (1,1) is *domain independent* iff for all $E \subseteq E'$, all $A, B \subseteq E$,

a. each $f \in D_E(A)(B)$ extends to an f' in $D_{E'}(A)(B)$ and
b. each $f' \in D_{E'}(A)(B)$ is an extension of some f in $D_E(A)(B)$.

Recall that to say that f' extends f above just says that it takes the same values at subsets of E as f does. And to say that English Dets are domain independent just says that they determine domain independent functionals. BLIK above fails DI since $\text{BLIK}\emptyset\emptyset = T$ when $E = \{a, b, c\}$ but $\text{BLIK}\emptyset\emptyset = F$ for $E' = \{a, b, c, d\}$. We note:

Theorem 11.6. *For each domain independent Det d and each model $\mathcal{M}$,*

a. $[\![d]\!]_{\mathcal{M}}$ is cardinal iff $[\![d]\!]_{\mathcal{M}}$ is both intersective and automorphism invariant, and
b. $[\![d]\!]_{\mathcal{M}}$ is co-cardinal iff $[\![d]\!]_{\mathcal{M}}$ is both co-intersective and automorphism invariant.

11.2.2 Conservativity

Conservativity[5] (CONS) is a local constraint, limiting the maps from $\mathcal{P}(E)$ into GQ_E which can be denotations of Dets. Specifically it requires that in evaluating $D(A)(B)$ we ignore the elements of B which do not lie in A. Formally,

[4]Van Benthem stated this condition slightly differently; calling it *Extension*.

[5]The term *conservative* dates from Keenan (1981); it was called *lives on* in Barwise and Cooper (1981) and *intersective* in Higginbotham and May (1981).

Definition 11.7. A function D from P_E into $[P_E \to X]$, X any set, is *conservative* iff for all $A, B, C \subseteq E$,

$$D(A)(B) = D(A)(C) \text{ whenever } A \cap B = A \cap C.$$

Theorem 11.7. *The set of conservative maps from $\mathcal{P}(E)$ into GQ_E is closed under the pointwise boolean operations.*

Theorem 11.8. *D from P_E to $[P_E \to X]$ is conservative iff for all $A, B \subseteq E$, $DAB = D(A)(A \cap B)$.*

Theorem 11.8 is the usual definition of Conservativity. The reader may use it to verify that the Dets in (23) are conservative by verifying that ($23x$,x') are logically equivalent.

(23)
- a. All swans are white.
- a'. All swans are swans and are white.
- b. Most bees buzz.
- b'. Most bees are bees that buzz.
- c. No dogs purr.
- c'. No dogs are dogs that purr.
- d. John's pupils work hard.
- d'. John's pupils are pupils and work hard.
- e. Which states have no taxes?
- e'. Which states are states with no taxes?
- f. Whose two bikes are in the yard?
- f'. Whose two bikes are bikes in the yard?
- g. Neither John's nor Bill's abstracts were accepted.
- g'. Neither John's nor Bill's abstracts were abstracts that were accepted.

The equivalences in (23) are felt as trivial because the predicate repeats information given by the noun. But surprisingly, Conservativity is a very strong constraint:

Theorem 11.9 (Keenan and Stavi (1986))**.** *The number of logically possible Det denotations is $|[P_E \to GQ_E]|$, namely $2^{4^{|E|}}$. The total number of these functions which are conservative is $2^{3^{|E|}}$.*

Thus in a two element domain E there are $2^{16} = 65,536$ maps from properties to GQs. Just $2^9 = 512$ are conservative! (And only 16 are (co-)intersective, of which 8 are (co-)cardinal). Functions like the Härtig quantifier H in (24), are not conservative: choose A and B to be disjoint singleton sets. Then $|A| = |B|$ but $|A| \neq |A \cap B|$.

(24) $H(A)(B) = T$ iff $|A| = |B|$.

Note that DI and CONS are independent: BLIK$_E$ above is CONS but not DI; the Härtig quantifier is DI but fails CONS in each E. It seems then that

(25) Natural Language Dets are both domain independent and conservative.

Ben-Shalom (2001) and Keenan and Westerståhl (1996) independently provide different single conditions which capture the combined effect of domain independence and conservativity.

Thus CONS is a strong condition, ruling out most maps from $\mathcal{P}(E)$ into GQ_E as possible Det denotations. But it still lets through many that lie outside the INT and CO-INT classes, e.g. the proportional Dets and the definite Dets. Are there others? Can we impose a local constraint stronger than CONS on possible Det denotations? If not, is there any reason to expect that the denotable Det functions should be just those admitted by CONS? Concerning stronger constraints Keenan and Stavi (1986) show that over any finite E any conservative function from $\mathcal{P}(E)$ into GQ_E can be denoted in the sense that we can construct an English expression (albeit a cumbersome one) which can be interpreted as that function. We also have a (more speculative) answer to the last question: Namely, consider that in general boolean compounds of intersective and co-intersective Dets lie in neither class. So the Dets italicized below are neither intersective nor co-intersective:

(26) a. *Some but not all* students read the Times.
b. *Either all or else not more than two* of the students will pass that exam.
c. *Either just two or three or else all but a couple of* students will pass that exam.

But these Dets are conservative since they are boolean compounds of conservative ones (by Theorem 11.7).

Theorem 11.10 (Keenan (1993)). *The set of conservative functions over E is the boolean closure of the intersective together with the co-intersective functions.*

That is, the set of functions obtained from INT and CO-INT by forming arbitrary meets, joins and complements is exactly the set of conservative functions. Recall our earlier remarks that the boolean connectives (*and*, *or*, *not*, *neither...nor...*) are not semantically tied to any particular category or denotation set, but rather express ways we have of conceiving of things—properties, relations, functions, etc.—regardless of what they are. Thus given that we need intersective and co-intersective functions as denotations for expressions we expect the class of CONS functions to be denotable just because we can form boolean compounds of expressions denoting (co)intersective functions.

There is a last property related to CONS and which distinguishes natural language quantifiers from their standard logical counterparts. Namely some Dets do not make critical use of the noun argument (the *restrictor*) to restrict the domain of quantification, but can be paraphrased by replacing the noun argument by one that denotes the whole domain, forming a new predicate by some boolean compound of the original noun and predicate argument. For example, as we saw, *Some As are Bs* says the same as *Some individuals are both As and Bs,* where in the second S we are quantifying over all elements of the domain, not just those in A, as in the first S. Indeed it is this conversion that is learned when we teach beginning logic students to represent *Some men are mortal* by "For some x, x is a man and x is mortal". The variable x here precisely ranges over the entire universe of objects under consideration. Similarly *All men are mortal* gets represented as "For all objects x, if x is a man then x is mortal".

Dets which admit of the elimination of the noun argument in this way will be called *sortally reducible*, and our key observation here is that English presents many Dets which are not sortally reducible. In fact the Dets that are sortally reducible are just the intersective and co-intersective ones. Let us more formally define:

Definition 11.8. D is *sortally reducible* iff for all $A, B \subseteq E$,

$$D(A)(B) = D(E)(\dots A \dots B \dots),$$

where "$(\dots A \dots B \dots)$" is a boolean function of A and B.

"Boolean functions" are ones definable solely in terms of the boolean connectives *and*, *or*, *not*, etc. In the formalism used here this just means that $(\dots A \dots B \dots)$ is defined in terms of set intersection ($\cap$), union ($\cup$) and complement ($\neg$). We use $A \to B$ to abbreviate $\neg A \cup B$.

Theorem 11.11.

a. *If D is intersective then D is sortally reducible, by*

$$DAB = D(E)(A \cap B).$$

This follows from the fact that $A \cap B = E \cap A \cap B$ plus the fact that D is intersective.

b. *If D is co-intersective then D is sortally reducible, by*

$$DAB = D(E)(A \to B).$$

Note that $E - (A \to B) = E \cap \neg(\neg A \cup B) = A \cap \neg B = A - B)$.

Theorem 11.12 (Keenan (1993))**.** *For D a function of type (1,1) over a domain E, D is sortally reducible iff either D is intersective or D is* *co-intersective.*

So, in general, proportionality Dets are not sortally reducible, they make essential use of the noun argument. Let us see that the reductions in Theorem 11.11 don't work for *most*. Theorem 11.11a doesn't, since *Most students are vegetarians* is not logically equivalent to *Most entities are both students and vegetarians*: given an E consisting of a hundred vegetarians only ten of whom are students, the first sentence is true and the second false. For the inadequacy of the if-then type reduction we show that (27b) does not entail (27a):

(27) a. Most students are vegans.
b. Most entities are such that if they are students then they are vegans.

Let $|E| = 100$ with just 15 students, no vegans. Then (27a) is false: none of the students are vegans. But (27b) is true: the if-then clause is satisfied by 85 of the 100 entities.

We have covered a variety of English Dets, all of the type that take two property denoting expressions as arguments to yield a truth value (or question meaning). But English arguably presents Dets of other types.

11.3 k-Place Dets

Following Keenan and Moss (1985) the italic expressions in (28) are Det_2s: they combine with two Ns to form a DP and semantically map pairs of sets to GQs.

(28) a. *More / Fewer* students *than* teachers came to the party.
b. *Fewer* students *than* teachers came to the party.
c. *Exactly / almost / not as many* students *as* teachers...
d. *More than ten times as many* students *as* teachers...
e. *How many more* students *than* teachers were arrested?
f. *The same number of* students *as* teachers...
g. *A larger / smaller number of* students *than* teachers...

(28a) gives the denotation of the Det_2 *more...than...*, with the N arguments in the easy-to-read order. Most of the other denotations in (28) are defined similarly. Read (29) as "More As than Bs have C iff the number of As with C exceeds the number of Bs with C".

(29) $(\text{MORE } A \text{ THAN } B)(C) = T$ iff $|A \cap C| > |B \cap C|$.

Exercise 11.4. On the pattern in (29) exhibit the denotations of the Det_2s below:

a. exactly as many... as...

b. more than twice as many... as...
c. no more... than...
d. the same number of...as...

Expressions such as *more students than teachers* which we treat as built from Det_2s are not usually treated as DPs in their own right. So it is worth noting that they share many distributional properties with standard DPs (Keenan (1987a)):

(30) a. John interviewed more men than women.
b. He sent flowers to more teachers than students.
c. She believes more students than teachers to have signed the petition.
d. More students than teachers are believed to have signed the petition.
e. More teachers than deans interviewed more men than women.
f. Almost all instructors and many more.
g. Ann knows more Danes than Germans but not more Danes than Swedes.

There is also an interesting semantic fact naturally represented on a view which treats *more...than...*, etc. as a Det_2 taking an ordered pair of nouns as arguments. Namely, modifiers of the pair naturally behave coordinate-wise applying to each property:

(31) More students than teachers at UCLA attended the rally
= More students at UCLA than teachers at UCLA attended the rally
$= (\text{AT UCLA})(p, q) = ((\text{AT UCLA})(p), (\text{AT UCLA})(q))$

Generalizing now, we can think of a Det_k as combining with a k-tuple of Ns to form a DP. Semantically it would map k-tuples of sets to GQs. Does English present Det_ks for $k > 2$? Perhaps. It is reasonable for example to treat *every...and...* as a k-place Det, all $k > 0$. (Think of *every* as the form it takes when $k = 1$). (32) interprets *every...and...* as a Det_3.

(32) a. Every man, woman and child jumped overboard.
b. $(\text{EVERY} \ldots \text{AND} \ldots)(A, B, C)(D) = \text{EVERY}(A \cup B \cup C)(D)$

The *and* in (32a) is not simple coordination, otherwise (32a) would mean that every object which was simultaneously a man, woman and child jumped overboard, which is empirically incorrect. The pattern whereby the value of a Det_k built from a Det_1 and *and* maps the n noun properties to whatever the Det_1 maps their union to, is quite general:

(33) For $D \in [P_E \to \text{GQ}_E]$, $(D \ldots \text{AND} \ldots)$ maps $(P_E)^k$ into GQ_E, all k, defined by:

$$(D \ldots \text{AND} \ldots)(A_1, \ldots, A_k) = D(\bigcup_{1 \le i \le k} A_i).$$

(34) a. About fifty men and women jumped overboard.
$(\text{ABOUT FIFTY}\ldots\text{AND}\ldots)(A, B)(D)$
$= (\text{ABOUT FIFTY})(A \cup B)(D)$

b. The sixty boys and girls laughed at the joke.
$(\text{THE SIXTY}\ldots\text{AND}\ldots)(A, B)(D) = (\text{THE SIXTY})(A \cup B)(D)$

c. Which boys and girls came to the party?
$(\text{WHICH}\ldots\text{AND}\ldots)(A)(B)(D) = \text{WHICH}(A \cup B)(D)$

The most convincing case that we have Det_2s in English comes from (28) as they do not lend themselves to a syntactic or semantic reduction to compounds of Dets of lesser arity.

The Det_2s in (28) are all cardinal (and thus intersective): the value of the DP they build at a predicate property B is decided by the cardinality of the intersection of B with each noun property. Defining conservativity, (co-)intersectivity, etc. for k-place Dets involves no real change from the unary case: a cardinal Det_k decides truth by checking the cardinality of the intersection of the predicate property with each of the k noun properties. A conservative Det_k checks the intersection of that property with each noun property, etc[6].

Comparative quantifiers (Beghelli (1992) Beghelli (1994) Smessaert (1996)) assume a variety of other forms in English as well. Those in (35) combine with a single noun property but two predicate properties to form a S.

(35) a. More students came early than left late.
b. Exactly as many students came early as left late.
c. The same number of students came early as left late.
d. The same students as came early left late.

[6]The extended definitions would proceed as follows:

Definition 11.9. For all k, all functions D from k-tuples of properties over E into $[P_E \to X]$,

a. D is *cardinal* iff for all k-tuples A, A' of properties and all properties B, B', if $|A_i \cap B| = |A'_i \cap B'|$, all $1 \le i \le k$, then $DAB = DA'B$.

b. D is *conservative* iff for all k-tuples $A = \langle A_1, \ldots, A_k \rangle$ and all properties B, B', if $A_i \cap B = A_i \cap B'$, all $1 \le i \le k$, then $DAB = DAB'$.

Define *intersectivity* for Det_ks by eliminating from (a) the cardinality signs; *co-cardinal* and *co-intersective* for Det_ks by replacing $A_i \cap B$ with $A_i - B$. For some additional types of Dets see Keenan and Westerståhl (1996)

Enriching our type notation we treat the Dets in (28) as of type ((1,1),1): they map a pair of noun properties to a function from properties to truth values. The Dets in (35) then perhaps have type (1,(1,1))), mapping a triple of properties a truth value. Note that the string *more students* in (35a) is not a DP and cannot be replaced by one.

(36) a. *All / Most / No students came early than left late.
b. *Exactly ten / Most students came early as left late.

The quantifiers of type ((1,1),1) and (1,(1,1)) we have considered are all cardinal in their type. We do not seem to find any co-cardinal two place Dets. There are however proportional Dets of type ((1,1),1) as in (37) and of type (1,(1,1)), as in (38).

(37) a. A greater/smaller percentage of students than (of) teachers signed the petition.
b. The same proportion of students as of teachers signed the petition.

(38) a. A greater / smaller percentage of students came early than left late.
b. The same percentage of students came early as left late.

And the Dets in (39) plausibly have type (1,1,1,1), or perhaps ((1,1),(1,1)). We only find natural cardinal examples:

(39) a. More students came early than teachers left late.
b. Just as many students came early as teachers left late.

Quantifiers in these high types have not, at time of writing, been subjected to any intensive study. Here we note simply two entailment paradigms which at least indicate that there is some logical behavior to study here. The first is due to Zuber (2008).

(40) a. More poets than linguists are vegetarians. (type ((1,1),1)
b. More vegetarians are poets than are linguists. (type (1,(1,1))

And in general,

(41) MORE A THAN A' ARE $B \leftrightarrow$ MORE B ARE A THAN ARE A'

(42) a. More poets than painters live in NY
b. $\equiv$ More poets who are not painters than painters who are not poets live in NY.

Exercise 11.5. Exhibit an informal model in which (a) is true and (b) is not.

a. The students who arrived early also left late.
b. The same students as arrived early left late.

11.4 Crossing the Frege Boundary

We have treated GQs as mapping P_1 denotations to S ($= \mathrm{P}_0$) denotations and we extended them to maps from P_{n+1} denotations to P_n denotations in such a way that their values at $n+1$-ary relations were determined by their values at the unary relations. This yielded the object narrow scope reading of Ss like *Some student praised every teacher*. The object wide scope reading was shown to be representable with the use of the lambda operator:

$$(\textsc{every teacher})\lambda x(\textsc{some student praised } x).$$

A suggestive aspect of our "extensions" approach is that it leads us to look for new types of DP denotations. Obviously there are many more maps from binary relations to unary relations over E ($|E| > 1$) than there are from unary relations to truth values. Can any of these new functions be denoted by expressions in English? In fact many can. Consider *himself* in (43a), interpreted as in (43b) and *everyone but himself* in (44a) interpreted as in (44b):

(43) a. Every poet admires himself.
b. $\textsc{self}(R) = \{b \in E \mid b \in R_b\}$

(44) a. No worker criticized everyone but himself.
b. $(\textsc{all but self})(R) = \{b \in E \mid R_b = E - \{b\}\}$

Recall that R_b is $\{y \in E \mid R(y)(b) = T\}$, the set of objects y which b bears the relation R to.

And provably these sorts of referentially dependent functions are not the extensions of any GQs to binary relations:

Theorem 11.13 (Keenan (1989))**.** *There is no GQ whose restriction to binary relations is* SELF *or* ALL BUT SELF *(for $|E| > 1$).*

The same holds for other DPs with ordinary pronominal forms bound to the subject, e.g. *his mother* in *Everyone$_i$ loves his$_i$ mother*.

Even more challenging are cases like (45) in which the pairs of italic expressions are felt to stand in some sort of mutual referential dependency relation.

(45) a. *Different people* like *different things*.
b. *Each student* answered a *different question* (on the exam).
c. John criticized Bill but *no one else* criticized *anyone else*.

And in fact (Keenan (1992)), there are no GQs F,G such that $F(G(\textsc{like}))$ always has the same truth value as (45a), all binary relations LIKE. But the pair (DIFFERENT PEOPLE, DIFFERENT THING) can be treated directly as a type (2) quantifier, one mapping binary relations to truth values. So such dependent pairs determine yet another type of quantification in natural language. In the same spirit, Moltmann (1996) notes inherently type (2) quantifiers like the exception constructions in (46).

(46) a. Every man danced with every woman except John with Mary.
b. No man danced with any woman except John with Mary.

11.5 A Classic Syntactic Problem

Linguists since Milsark (1977) have puzzled over which DPs occur naturally in Existential There (ET) contexts, as in (47) and (48):

(47) a. There are at most three undergraduate students in my logic class.
b. Isn't there at least one student who objects to that?
c. Aren't there more than five students enrolled in the course?
d. There were more students than teachers arrested at the demonstration.
e. Just how many students were there at the party?
f. Were there the same number of students as teachers at the party?
g. There weren't as many students as teachers at the lecture.

The examples in (47) are all built from cardinal Dets. Of note is that cardinal Det_2s, not traditionally considered in this (or any) context, build DPs fully acceptable in ET contexts.

Exercise 11.6. On the basis of examples like those in (47)

a. Exhibit four structurally distinct boolean compounds of DPs acceptable in ET contexts and show that they themselves are also acceptable in ET contexts.
b. Exhibit some ungrammatical boolean compounds of DPs built from cardinal Dets. Can you suggest any regularities limiting the formation of such DPs?
c. Can you find any DPs acceptable in ET contexts whose boolean compounds are grammatical in general but not acceptable in ET contexts? If not this argues that ET contexts do not impose any special restrictions on boolean compounding.

There are many pragmatic issues involved with judgments of acceptability of DPs in ET constructions, but a good (but not perfect, see below) approximation to a proper characterization of the acceptable DPs here is that they are just those built from intersective Dets (Det_1s or Det_2s) and their boolean compounds. So, like the definite plural

DPs, we characterize a class of DPs in terms of the Dets that build them. The Dets in (47) are cardinal, those in (48) are intersective but not cardinal. Those in (49) are not intersective and the resulting Ss are marginal to bad:

(48) a. Aren't there *as many male as female students* in the class?
b. There was *no student but John* in the building.
c. Just *which students* were there at the party anyway?
d. There were *only two students besides John* at the lecture.

(49) a. *There are *most students* in my logic class.
b. *Isn't there *the student* who objects to that?
c. ??Aren't there *seven out of ten students* enrolled in your course?
d. *Isn't there *every student* who gave a talk at the conference?
e. *Was there *neither student* arrested at the demonstration?
f. ??*Which of the two students* were there at the demonstration?

See Reuland and ter Meulen (1987) for several articles discussing DPs in Existential There contexts. Keenan (2003) is a recent proposal which entails the intersectivity claim above. Peters and Westerståhl (2006) (pp. 214–238) is a more recent in depth review of the literature and notes (at least) two problems with the intersectivity thesis above. Namely, partitives, as in (50) and possessives, as in (51).

(50) I believe that there are at least two of the five supervisors that favor that bill.

Treating the Det in (50) as *at least two of the five* it will fail to be intersective as it is not symmetric: setting

$$D = (\text{AT LEAST TWO OF THE FIVE}),$$

we have that $D(A)(B)$ is true if $|A| = 5$, $|B| = 10$ and $|A \cap B| \geq 2$. But then $D(B)(A)$ is false since $|B| \neq 5$. And seemingly all options of analysis must enforce an asymmetry between A and B with $|A|$ required to be 5 and $|B|$ not so required.

(51) a. There is *some neighbor's dog* that barks incessantly.
b. *There is *each neighbor's dog* that barks incessantly.

Possessives are difficult both syntactically and semantically. The most thorough semantic treatment is in fact that given in Chapter 7 of Peters and Westerståhl (2006). Possessive Dets may fail to be symmetric hence they fail to be intersective: it may be true that John's doctor is a lawyer but false that John's lawyer is a doctor. Possibly possessive Dets are (sometimes) properly intensional, like *too many* and *not enough* noted in footnote 1. But that won't explain the judgments in (51a,b) as the sense of possession is the same in the two cases.

We conclude with a brief look at some tantalizing but less well understood instances of A(dverbial)-quantification in English, as opposed to the D(eterminer)-quantification (Partee (1994)) we have been considering.

11.6 Adverbial Quantification

A(dverbial)-quantification is expressible with independent adverbs or PPs: Lewis (1975), Heim (1982), de Swart (1996, 1994).

(52) a. John *always / usually / often / occasionally / rarely / never* trains in the park.
b. John took his driver's exam *twice / (more than) three times*.
c. Mary brushes her teeth *every day / twice a day*.

There is a striking semantic correspondence between the adverbial quantifiers underlined above and the D-Dets presented earlier. *Always* corresponds to *all*, *never* to *no*, *twice* to *two*, *usually* to *most*, *occasionally / sometimes* to *some*, and *often* and *rarely* to *many* and *few*. Similarly Bittner (1995) lists pairs of A- and D- quantifiers (translating *always*, *mostly*, *often*, *sometimes*) in Greenlandic Eskimo formed from the same root but differing in adverbial vs nominal morphology. And Evans (1995) lists pairs of semantically similar D- and A-quantifiers in Mayali (Australia).

In general what we quantify over in the D-cases is given by the Ns the Det combines with. But in A-quantification there is often no such clear constituent, and precisely what we are quantifying over is less clear. One influential approach to A-quantification follows up on the unselective binding approach in Lewis (1975). Examples that illustrate A-quantifiers best and which seem empirically most adequate are ones lacking independent D-quantifiers. (53) is from (Peters and Westerståhl, 2006, pg. 352).

(53) a. Men are usually taller than women.
b. $\text{MOST}_2(\{\langle x,y\rangle \mid x \in \text{MAN} \;\&\; y \in \text{WOMAN}\}, \{\langle x,y\rangle \mid x \text{ TALLER } y\}) = T$
c. iff
$$\frac{|(\text{MAN} \times \text{WOMAN}) \cap \text{TALLER}|}{|(\text{MAN} \times \text{WOMAN})|} > \frac{1}{2}$$

On this interpretation (53a) is true iff the proportion of (man,woman) pairs in the *taller than* relation is more than half the (man,woman) pairs. So MOST_2 is of type (2,2), mapping a pair of binary relations $\text{MAN} \times \text{WOMAN}$ and TALLER as arguments to truth values, as it does in simpler cases like *Most colleagues are friends.* Its semantics is that of MOST given earlier, but now the sets it intersects and compares cardinalities of are sets of ordered pairs. In general for D any of our original Det functions, D_k, the k-resumption of D, is that function like D except that its arguments are k-ary relations. Peters and Westerståhl (2006) call this lifting operation *resumption.* It is one way A-quantifiers are characterized in terms of D-quantifiers. Thus it is immediate how to interpret the Ss differing from (53a) by replacing *usually* with *always*, *sometimes*, and *never*.

Now, to what extent can we represent A-quantification by resumption? We don't know at time of writing, but one more case that has been widely treated as unselective binding (resumption) is biclausal constructions built from *when/if* clauses and generic or indefinite DPs (constructed with the indefinite article *a/an*), as (54). See Kratzer (1995).

(54) a. (Always) when a linguist buys a book he reads its bibliography first.
b. $\text{ALL}_2(R, S)$, where
$R = \{\langle x, y\rangle \mid x \in \text{LINGUIST}, y \in \text{BOOK}, x \text{ BUY } y\}$ and
$S = \{\langle x, y\rangle \mid x \text{ READ } y\text{'s BIBLIOGRAPHY FIRST}\})$
c. $= T$ iff $R \subseteq S$

ALL_2 is just ALL with binary not unary relation arguments. (54c) says that (54a) is true iff for all linguists x, all books y, if x buys y then x reads y's bibliography first, which seems right. *Always* can be replaced by *Sometimes*, *Never*, and *Not always*, interpreted by SOME_2, NO_2, and $\neg\text{ALL}_2$ with the intuitively correct truth conditions. However further extensions to Ss in which A- and D-quantification interact have not been successful. A much studied example is the Geach "donkey" sentence (Geach (1962)), as in (39a) with it anaphoric to *donkey.* Kamp (1981) and Heim (1982) among others have tried to interpret it with resumptive quantification as in (55b).

(55) a. Every farmer who owns a donkey beats it.
b. $\text{ALL}_2(\{\langle x, y\rangle \mid x \in \text{FARMER}, y \in \text{DONKEY} \;\&\; x \text{ OWN } y\},$
$\{\langle x, y\rangle \mid x \text{ BEAT } y\})$

This yields the "strong" interpretation on which every farmer who owns a donkey beats every donkey he owns. Several linguists either accept this interpretation or at least feel that it is the closest clear statement of the truth conditions of (55a). But most choices of initial quantifier do not yield correct resumptive interpretations. Kanazawa (1994) notes that the resumptive reading of

(56) a. At least two farmers who own a donkey beat it.
b. Most farmers who own a donkey beat it.

(56a) would, incorrectly, be true in a model in which there are just two farmers, one owns one donkey and doesn't beat it, the other owns two and beats them both. Rooth (1987) notes the comparable problem for (56b) in which say all but one of ten farmers owns just one donkey and beats it, but the last farmer owns 100 donkeys and doesn't beat any of them. This problem is now called the *proportion* problem, a misnomer since, per Kanazawa, it arises with non-proportional Dets like at least *two* as well. Indeed Peters and Westerståhl (2006) attribute to van der Does and van Eijck (1996) the claim that only *all*, *some* and their complements don't lead to a proportion problem. In addition Chierchia (1992) cites cases in which Ss like (57a) get a "weak" or "existential" reading, not a universal one.

(57) a. Everyone who has a credit card will pay his bill with it. (Cooper (1979))
b. Everyone who has a dime will put it in the meter. (Pelletier and Schubert (1989))

Evans (1977), Cooper (1979) and, in a different way, Heim (1990), try to handle the "dangling" *it* in donkey Ss with E-type pronouns, in effect replacing it by a full DP such as *the donkey he owns*, where *he* refers back to *farmer*. But the results are less than satisfactory when some farmers own more than one donkey. From our perspective these proposals do not so much invoke new quantifiers as establish the scope of familiar quantifiers. Later proposals by Groenendijk and Stokhoff (1991), Chierchia (1992), Kanazawa (1994) and de Swart (1994) invoke dynamic logic in which natural language expressions are represented in a logical language and variables not in the syntactic scope (the c-command domain) of a vbo can nonetheless be bound by it.

So far we have not considered the domain of the resumptive quantifier in a systematic way. In (53a) the two Ns *man* and *woman* are part of different DP constituents, yet the domain of the quantifier is the cross product of their denotations. In (54a) it was the subordinate *when* clause in which we abstracted twice to form a binary relation denoting expression. Now returning to our initial example, repeated as (58a), we don't find naturally constructable binary relations of the relevant sorts. Rather, following de Swart (1996), it seems that we are comparing the "times" (or "occasions") John trains with the times he trains in the park.

(58) a. John always / usually / ... trains in the park.
b. ALL($\{t \mid$ John trains at $t\}, \{t \mid$ John trains in the park at $t\}$)

So the Ss in (58a) compare the set of times John trains with the set of times he trains in the park. ALWAYS says that the first set is included in the second; NEVER says they are disjoint; SOMETIMES says they are not; USUALLY says that the set of times he trains in the park number more than half of the number of times that he trains, etc. So here A-quantification is handled as D-quantification over times. This approach is not unnatural given A-quantifiers which overtly mention times as *sometimes*, *five times*, *most of the time*, *from time to time*. Moreover it enables us to test whether the properties we adduced for D-quantifiers extend to their corresponding A-ones. And several do, as de Swart (1996) shows.

The cases in (58) are trivially Conservative. For any A-quantifier Q,

(59) Q(TRAIN)(TRAIN IN THE PARK)
$= Q$(TRAIN)(TRAIN $\cap$ TRAIN IN THE PARK).

They are also Domain Independent: if more times are added to the model but the two arguments of an A-Quantifier are unchanged then the value Q assigns them is unchanged. Further some A-Quantifiers are intersective: SOMETIMES, NEVER; some are co-intersective: ALWAYS, WITH JUST TWO EXCEPTIONS, (60a); and some properly proportional, (60b): USUALLY, MORE THAN TWO THIRDS OF THE TIME. (As with D-quantifiers the notion of proportion is clearest when the arguments are finite and non-empty).

(60) a. With two exceptions, John has always voted for a Democrat for President.
b. More than two thirds of the time when John prayed for rain it rained.

de Swart (1996) also handles some subordinate temporal clauses with *before* and *after* which are not mere place holders for quantificational domains in the way that *when* and *if* clauses may be.

(61) a. Paul *always* takes a shower just before he goes to bed.
b. Paul *never* exercises immediately after he has had dinner.

(61a) says that the times just before Paul goes to bed are all among those when he takes a shower. (61b) says that the times immediately after he has had dinner are disjoint from the times he exercises. *Usually*, *always*, *sometimes* and *never* are interpretable by their corresponding D-Det. Using *when* as an argument slot definer we see that the A-quantifiers above have the monotonicity properties of their D-counterparts. Like *all*, *always* is increasing on its second argument, decreasing on its first, so the inferences in (62) are valid and npi's are licensed in the first argument but not the second, (63).

(62) a. Always when John travels he reads a book.
b. $\Longrightarrow$ Always when John travels he reads something.
c. $\Longrightarrow$ Always when John travels by train he reads a book.

(63) a. Always when *anyone* travels he reads a book.
b. *Always when John travels he reads *any* book.

Lewis (1975) cautioned against a "times" approach noting that donkey Ss refer more to a continuing state than an event, and Ss like *A quadratic equation usually has two different solutions* lack a time coordinate altogether. This is certainly true, though it leaves unexplained why it is natural to use the temporal metaphor in discussing mathematical Ss. A logician might say that a set of sentences is semantically consistent if they can be simultaneously true. Lewis himself notes that Russell and Whitehead (1910) use *always* and *sometimes* in explaining their introduction of the now standard universal and existential quantifier: $(x).\varphi x$ means φx *always*, $(\exists x).\varphi x$ means φx *sometimes*. It would not seem problematic to interpret Ss as functions taking "abstract times" as arguments, with truly "timeless" Ss denoting constant functions, as with vacuous quantification generally. Artstein (2005), building on Pratt and Francez (2001) treats *before* and *after* phrases (*after the meeting*, *after John left*) as temporal generalized quantifiers—they map properties of time intervals to $\{T, F\}$.

11.7 Concluding Remarks

D-quantification over count domains is the best understood type of quantification in natural language. Our knowledge in this domain has grown enormously beginning in the 1980s. And we see that it proves helpful in understanding mass and A-quantification, both areas currently being researched and in which many empirical and conceptual issues remain unexplored, even unformulated.

11.8 Historical Background

Quantification has been a major topic of both syntactic and semantic investigation in natural language for some time. In addition to articles cited in the body of this chapter, some good collections or overview articles are: van Benthem and ter Meulen (1985), van Benthem (1984a), Reuland and ter Meulen (1987), van der Does and van Eijck (1996), Gärdenfors (1987), Kanazawa and Piñón (1994). Westerståhl (1989) overviews of this work up to 1987; Keenan and Westerståhl (1996) cover the later work; Keenan (1996) and Keenan (2008) are more linguistically oriented overviews. Peters and Westerståhl (2006) is the most comprehensive and in depth work to date. More purely mathematical work stimulated in part by this activity is: van Benthem (1984b), Westerståhl (1985), Keenan (1993), Kolaitis and Väänänen (1995), and the recent collection Krynicki et al. (1995). Szabolcsi (1997) is an excellent source for issues concerning scope, branching, and distributivity in natural language. Unselective binding and adverbial quantification are being investigated by several people. See for example Partee (1985) and de Swart (1994). Verkuyl and van der Does (1996), Lønning (1985), Winter (1998) discuss plurals and collectives. Doetjes and Honcoop (1997) and Krifka (1989, 1990) are good sources for work on event quantification. See Groenendijk and Stokhoff (1985) and Gutierrez-Rexach (1997) for recent work on questions and quantification.

11.9 Appendix: Some types of English Determiners

We present a variety of subclasses of Dets below. The classes are only informally indicated and are not exclusive some Dets appear in several classes. The intent here is to give the reader some idea of the syntactic and semantic diversity of English Dets. As the reader will see, we are generous with what we call a Det, since generalizations that we make about the entire class will remain valid if some of our Dets are syntactically reanalyzed in other ways. Had we chosen a too narrow class initially some of our generalizations might be vitiated simply by bringing up new Dets not considered.

(64) **$\mathbf{Det_1}$'s:** These are Dets which combine with one (possibly complex) Noun to form a P_0/P_1—a generalized quantifier denoting expression. These include the Dets we have already considered, like *every*, *some*, *no*, etc. and many others:

Lexical Dets: every, each, all, some, a, no, several, neither, most, the, both, this, my, these, John's, ten, few, many, a few, a dozen,

Cardinal Dets: exactly ten, approximately/more than/fewer than/at most/only ten, infinitely many, two dozen, between five and ten, just finitely many, an even/odd/large number of

Approximative Dets: almost all/no, practically no, approximately/about/nearly/around fifty, a hundred plus or minus ten

Definite Dets: the, that, this, these, my, his, John's, the ten, these ten, John's ten

Exception Dets: all but ten, all but at most ten, every...but John, no...but Mary

Bounding Dets: exactly ten, between five and ten, most but not all, exactly half the, just one...in ten, only SOME (=some but not all; upper case = contrastive stress), just the LIBERAL, only JOHN'S

Possessive Dets: my, John's, no student's, either John's or Mary's, neither John's nor Mary's

Comparative Possessives: more of Mary's than of Ann's (articles were cited)

Value Judgment Dets: too many, too few, a few too many, (not) enough, surprisingly few, ?many, ?few, more ... than we expected

Proportionality Dets: most, two out of three, (not) one ...in ten, less than half the (these, John's), exactly/more than/about/nearly half the, (more than) a third of the, ten per cent of the, every second

Partitive Dets: most/two/none/only some of the ten / of John's, more of John's than of Mary's, not more than two of the ten

Negated Dets: not every, not all, not a (single), not more than ten, not more than half, not very many, not quite enough, not over a hundred, not one of John's, not even two per cent of the

Conjoined Dets: at least two but not more than ten, most but not all, either fewer than ten or else more than a hundred, both John's and Mary's, at least a third and at most two thirds of the, neither fewer than ten nor more than a hundred

Adjectively Restricted Dets: John's biggest, more male than female, most male and all female, the last...John visited, the first...to set foot on the Moon, the easiest...to clean, whatever...are in the cupboard, the same...who came early

Logical Dets: every, no, most, the two, all but two, exactly two, most but not all, just two of the ten, not more than ten, at least two and not more than ten, seven out of ten, not one...in ten

(65) **Det$_2$s:** These combine with two Ns to form a DP, as in *more students than teachers (came to the party).*

Cardinal comparatives: more...than..., fewer...than..., exactly as many...as..., five more...than..., twice as many ...as..., the same number of...as...

Coordinate extensions: every...and..., no...or...., the more than twenty...and..., some...and...

The three dots above indicate the locus of the N arguments. E.g. in *not one student in ten* we treat *not one...in ten* as a discontinuous Det. In general we have two reasons for positing discontinuous analyses. One, often the N+postnominal material, such as *student in ten* in *not one student in ten*, has no reasonable interpretation, and so is not naturally treated as a constituent (which, by Compositionality, should be interpreted).

And two, the presence of the postnominal and prenominal material may fail to be independent. If in *not one student in ten* we treated *student in ten* as a N which combines with the Det *not one*, how would we block **the/this/John's student in ten*? So there are sensible reasons for treating the complex expressions above as Dets, though this proposal is not without problems of its own (Lappin (1996), Rothstein (1988)) and very possibly some of our cases will find a non-discontinuous analysis (see von Fintel (1993), Lappin (1996), Moltmann (1996) on exception Dets)

References

Abiteboul, Serge, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases*. Addison-Wesley.

Abney, Steven and Mark Johnson. 1991. Memory requirements and local ambiguities for parsing strategies. *Journal of Psycholinguistic Research* 20(3):233–250.

Aissen, Judith L. 1987. *Tzotzil Clause Structure*. Dordrecht: D. Reidel Pub. Co.

Ajdukievicz, Kasimierz. 1967. Syntactic connexion in English. In S. McCall, ed., *Polish Logic 1920–1939*, pages 207–231. Oxford, UK: Oxford University Press.

Artstein, Ron. 2005. Quantificational arguments in temporal adjunct clauses. *Linguistics and Philosophy* 28:541–597.

Büring, Daniel. 2005. *Binding Theory*. Cambridge, UK: Cambridge University Press.

Babyonyshev, Maria and Edward Gibson. 1999. The complexity of nested structures in Japanese. *Language* 75(3):423–50.

Bach, Emmon, Eloise Jelinek, Angelika Kratzer, and Barbara Partee, eds. 1995. *Quantification in Natural Language*. Dordrecht: Kluwer.

Baker, Mark C. 2001. *The Atoms of Language*. New York: Basic Books.

Bar-Hillel, Yehoshua, Chaim Gaifman, and E. Shamir. 1960. On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel* 9. reprinted in Bar-Hillel Language and Information. Addison-Wesley Pub. Co. Reading, Mass. 1964.

Barker, Chris and Pauline Jacobson. 2007. *Direct Compositionality*. Oxford University Press.

Barlow, M. and C.A. Ferguson. 1988. *Agreement in Natural Language*. CSLI.

Barwise, Jon. 1979. On branching quantifiers in English. *Journal of Philosophical Logic* 8:47–80.

Barwise, Jon and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4:159–219.

Beghelli, Filippo. 1992. Comparative quantifiers. *Proc. of the Eighth Amsterdam Colloquium* .

Beghelli, Filippo. 1994. Structured quantifiers. In Kanazawa and Piñón (1994), pages 119–145.

Beghelli, Filippo, Dorit Ben-Shalom, and Anna Szabolcsi. 1997. Variation, distributivity, and the illusion of branching. In A. Szabolcsi, ed., *Ways of Scope Taking*. Dordrecht: Kluwer.

Ben-Shalom, Dorit. 2001. One connection between standard invariance conditions on modal formulas and generalized quantifiers. *Jornal of Logic, Language, and Information* 10:1–6.

Bernardi, Raffaella. 2002. *Reasoning with Polarity in Categorial Type Logic*. Ph.D. thesis, University of Utrecht.

Bittner, Maria. 1995. Quantification in Eskimo: A challenge for compositional semantics. In Bach et al. (1995).

Blevins, James. 1994. Derived constituent order in unbounded dependency constructions. *J. Linguistics* 30:349–409.

Blust, R. 2001. Thao triplication. *Oceanic Linguistics* 40(2):324–335.

Boole, George. 1854. *The Laws of Thought*. Macmillan. Reprinted with corrections, Dover Publications, New York NY, 1958.

Boolos, George. 1981. For every A there is a B. *Linguistic Inquiry* 12:465–466.

Boolos, George and Richard Jeffrey. 1980. *Computability and Logic*. Cambridge University Press, 2nd edn.

Büchi, Richard J. 1989. *Finite Automota, Their Algebras and Grammars*. Springer Verlag.

Bunch, B. 1989. *Reality's Mirror*. John Wiley & Sons.

Bybee, J., R. Perkins, and W. Pagliuca. 1994. *The Evolution of Grammar*. University of Chicago Press.

Chierchia, Gennaro. 1992. Anaphora and dynamic binding. *Linguistics and Philosophy* 15:111–183.

Chierchia, Gennaro. 2004. Scalar implicatures, polarity phenomena and the syntax/pragmatics interface. In A. Belletti, ed., *Structures and Beyond*, vol. 3, pages 39–104. Oxford University Press.

Chomsky, Noam. 1996. *The Minimalist Program*. Cambridge, MA: The MIT Press.

Chomsky, Noam and Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry* 8:425–504.

Chomsky, Noam and George A. Miller. 1963. Introduction to the formal analysis of natural languages. In R. Luce, R. Bush, and E. Galanter, eds., *The Handbook of Mathematical Psychology*, vol. II, chap. 11. Wiley.

Church, Alonzo. 1936. A note on the Entscheidungsproblem. *Journal of Symbolic Logic* pages 40–41.

Church, Kenneth W. 1980. *On Memory Limitations in Natural Language Processing*. Master's thesis, MIT.

Cole, P. and G. Hermon. 2008. Vp raising in a vos language. *Syntax* 11(2):144–197.

Comrie, Bernard. 1985. Derivational morphology. In T. Shopen, ed., *Language Typology and Syntactic Description*, vol. 3, pages 309–349. Cambridge, UK: Cambridge University Press.

Comrie, Bernard. 2000. Valency-changing derivations in Tsez. In R. Dixon and A. Aikhenvald, eds., *Changing Valency: Case Studies in Transitivity*, pages 360–375. Cambridge, UK: Cambridge University Press.

Cooper, Robin. 1979. The interpretation of pronouns. In F. Heny and H. Schnelle, eds., *Syntac and Semantics*, vol. 10. Academic Press.

Cooper, Robin. 1982. Binding in wholewheat syntax (unenriched with inaudibilia). In P. Jacobson and G. Pullum, eds., *The Nature of Syntactic Representation*, pages 59–77. Reidel.

Corbett, Greville. 1991. *Gender*. Cambridge, England: Cambridge University Press.

Cotton, A., F.İ990. *Chemical Applications of Group Theory*. John Wiley & Sons.

Craig, William. 1957. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic* 22(3):269–285.

de Roeck et al., Anne. 1982. A myth about centre-embedding. *Lingua* 58:327–340.

de Swart, Henriette. 1994. (in)definites and genericity. In Kanazawa and Piñón (1994), pages 171–195.

de Swart, Henriette. 1996. Quantification over time. In van der Does and van Eijck (1996), pages 311–337.

Doetjes, Jenny and Martin Honcoop. 1997. The semantics of event-related readings: a case for pair quantification. In Szabolcsi (1997), pages 263–297.

Dowty, David. 1982. Derivational morphology. *Linguistics and Philosophy* 5:23–58.

Dryer, Matthew. 2007. Word order. In T. Shopen, ed., *Language Typology and Syntactic Description*. Cambridge University Press, 2nd edn.

Emmorey, K. 1984. The intonation system of toba batak. In *Studies in the Structure of Toba Batak*, UCLA Working Papers on Linguistics. UCLA.

Enderton, Herbert B. 1972. *A Mathematical Introduction to Logic*. NY: Academic Press.

Enderton, Herbert B. 1977. *Elements of set theory*. New York-London: Academic Press.

Enderton, Herbert B. 1985. Elements of recursion theory. In J. Barwise, ed., *The Handbook of Mathematical Logic*, pages 527–567. Elsevier.

Evans, Gareth. 1977. Pronouns, quantifiers, and relative clauses. *Canadian Journal of Philosophy* 7:467–536.

Evans, Nick. 1995. A-quantifiers and scope in Mayali. In Bach et al. (1995).

Fauconnier, Gilles. 1975. Do quantifiers branch? *Linguistic Inquiry* 6:555–578.

Flum, J. 1975. Characterizing logics. In J. Barwise and S. Feferman, eds., *Model-Theoretic Logics*, chap. 3, pages 77–121. Springer-Verlag.

Gärdenfors, Peter, ed. 1987. *Generalized Quantifiers: Linguistic and Logical Approaches*. Dordrecht: Reidel.

Gabbay, Dov and Julius Moravcsik. 1974. Branching quantifiers, English, and Montague grammar. *Theoretical Linguistics* 1:141–157.

Gardner, M. 2005. *The New Ambidextrous Universe*. Dover, 3rd edn.

Geach, Peter. 1962. *Reference and Generality*. Cornell University Press.

Gil, David. 1993. Nominal and verbal quantification. *Sprachtypologie und Universalienforschung* 46(4):275–317.

Givon, T. 1971. Historical syntax and synchronic morphology: an archaeologist's field trip. *CLS* 7:394–415.

Grädel, Erich, Phokion Kolaitis, and Moshe Vardi. 1997. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic* 3(1):53–69.

Groenendijk, Jeroen and Martin Stokhoff. 1985. Questions. In van Benthem and ter Meulen (1985), pages 1055–1125.

Groenendijk, Jeroen and Martin Stokhoff. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14:39–100.

Guenthner, Franz and J. Hoepelmann. 1974. A note on the representation of "branching quantifiers". *Theoretical Linguistics* 1:285–291.

Gutierrez-Rexach, Javier. 1997. Questions and generalized quantifers. In Szabolcsi (1997), pages 409–453.

Halmos, Paul. 1974. *Naive set theory*. Undergraduate Texts in Mathematics. New York-Heidelberg: Academic Press.

Harel, David. 1987. *Algorithmics: The spirit of computing*. UK: Addison-Wesley.

Heim, Irene. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, UMass Amherst.

Heim, Irene. 1990. E-type pronouns and donkey anaphora. *Linguistics and Philosophy* 13:137–177.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

Heine, B. and T. Kuteva. 2002. *World Lexicon of Grammaticalization*. Cambridge University Press.

Heine, B. and M. Reh. 1984. *Grammatical Categories in African Languages*. Hamburg: Helmut Buske.

Higginbotham, James. 1994. Mass and count quantifiers. *Linguistics and Philosophy* 17(5):447–480.

Higginbotham, James and Robert May. 1981. Questions, quantifiers, and crossing. *Linguistic Review* 1:41–79.

Hindley, J. Roger and J. P. Seldin. 1986. *Introduction to Combinators and the λ-Calculus*. Cambridge University Press. London Mathematical Society Student Texts.

Hintika, Jaako. 1974. Quantifiers vs. quantification theory. *Linguistic Inquiry* 5:153–177.

Hopper, P.J. 1991. On some principles of grammaticalization. In E. Traugott and B. Heine, eds., *Grammaticalization Vol. 1*. Amsterdam: John Benjamins.

Hopper, P.J. and E.C. Traugott. 1993. *Grammaticalization*. Cambridge University Press.

Horn, Laurence R. 1989. *Natural History of Negation*. Chicago: University of Chicago Press.

Huck, Geoffrey and Almerindo Ojeda. 1987. *Discontinuous Constituency: Syntax and Semantics*. New York: Academic Press.

Kahr, A. S., Edward F. Moore, and Hao Wang. 1962. Entscheidsproblem reduced to the $\forall\exists\forall$ case. *Proceedings of the National Academy of Sciences* 48(365–377).

Kamp, Hans. 1981. A theory of truth and semantic representation. In J. Groenendijk et. al, ed., *Formal Methods in the Study of Language*, pages 277–322. Mathematisch Centrum.

Kanazawa, Makoto. 1994. Weak vs. strong readings of donkey sentences and monotonicity inferences in a dynamic setting. *Linguistics and Philosophy* 17(2):109–159.

Kanazawa, Makoto and Chris Piñón, eds. 1994. *Dynamics, Polarity, and Quantification*. Lecture Notes 48. CSLI.

Kandybowicz, J. 2008. *The Grammar of Repetition*. John Benjamins.

Kayne, R. 1984. *Connectedness and Binary Branching*. Foris.

Keenan, Edward L. 1979. On surface form and logical form. *Studies in Linguistic Science* 8(2).

Keenan, Edward L. 1981. A boolean approach to semantics. In J. G. et al., ed., *Formal Methods in the Study of Language*. Math. Centre, University of Amsterdam.

Keenan, Edward L. 1987a. Multiply-headed NPs. *Linguistic Inquiry* 18(3):481–490.

Keenan, Edward L. 1987b. A semantic definition of 'indefinite NP'. In *The Representation of (In)Definiteness*. Cambridge, MA: MIT Press.

Keenan, Edward L. 1989. Semantic case theory. In *Semantics and Contextual Expression*, page 3357. Dordrecht: Foris.

Keenan, Edward L. 1992. Beyond the Frege boundary. *Linguistics and Philosophy* 15:199–221.

Keenan, Edward L. 1993. Natural language, sortal reducibility and generalized quantifiers. *Journal of Symbolic Logic* 58(1):314–325.

Keenan, Edward L. 1995. Predicate-argument structure in Malagasy. In C. Burgess, K. Dziwirek, and D. Gerdts, eds., *Grammatical Relations: Theoretical Approaches to Empirical Questions*, pages 171–217. Stanford: CSLA.

Keenan, Edward L. 1996. The semantics of determiners. In S. Lappin, ed., *The Handbook of Contemporary Semantic Theory*. Blackwell.

Keenan, Edward L. 2003. The definiteness effect: sematics or pragmatics? *Natural Language Semantics* 11(2):187–216.

Keenan, Edward L. 2007. On the denotations of anaphors. *Research on Language and Computation* 5(1):5–17.

Keenan, Edward L. 2008. Quantifiers. In K. V. Heusinger, C. Maienborn, and P. Portner, eds., *The Handbook of Semantics*. Mouton de Gruyter, 2nd edn.

Keenan, Edward L. 2009. Voice determines co-argument anaphora in west austronesian. *Proceedings of the XVIth Annual Meetings of the Austronesian Formal Linstuics Association* .

Keenan, Edward L. and Matthew S. Dryer. 2007. Passive in the world's languages. In T. Shopen, ed., *Language Typology and Syntactic Description*, vol. 1, pages 325–362. Cambridge, UK: Cambridge University Pres.

Keenan, Edward L. and Leonard Faltz. 1985. *Boolean Semantics for Natural Language*. Reidel.

Keenan, Edward L. and Lawrence S. Moss. 1985. Generalized quantifiers and the expressive power of natural language. In *Generalized quantifiers in Natural Language*. Foris.

Keenan, Edward L. and Edward P. Stabler. 2003. *Bare Grammar*. Stanford: CSLI.

Keenan, Edward L. and Jonathan Stavi. 1986. A demantic characterization of natural language determiners. *Linguistics and Philosophy* 9:253–326.

Keenan, Edward L. and Dag Westerståhl. 1996. Generalized quantifiers in linguistics and logic. In *Handbook of Language and Logic*, page 837893. Elsevier.

Kimenyi, A. 1980. *A Relational Grammar of Kinyarwanda*. UC Press.

Klima, Edward. 1964. Negation in English. In *The Structure of Language*. Prentice-Hall.

Kobele, G. 2006. *Generating Copies: An investigation into Structural Identity in Language and Grammar*. Ph.D. thesis, UCLA.

Kolaitis, Phokion and J. Väänänen. 1995. Generalized quantifiers and pebble games on finite structures. *Annals of Pure and Applied Logic* 74:23–75.

Kolaitis, Phokion G. 2006. On the expressive power of logics on finite models. In E. Grädel et. al, ed., *Finite model theory and its applications*, pages 27–123. Springer.

Kratzer, Angelika. 1995. Stage-level and individual level predicates. In G. Carlson and F. J. Pelletier, eds., *The Generic Book*, pages 125–175. University of Chicago Press.

Krifka, Manfred. 1989. Nominal reference, temporal constitution and quantification in event semantics. In R. Bartsch, J. van Benthem, and P. van Emde Boas, eds., *Semantics and Contextual Expression*, pages 75–117. Dordrecht: Foris.

Krifka, Manfred. 1990. Four thousand ships passed through the lock. *Lingustics and Philosophy* 15(5):487–520.

Krynicki, Michal, Marcin Mostowski, and Leslaw Szczerba, eds. 1995. *Quantifiers: Logics, Models and Computation*. Dordrecht: Kluwer.

Ladusaw, William. 1983. Logical form and conditions on grammaticality. *Linguistics and Philosophy* 6:389–422.

Lambek, Joachim. 1958. The mathematics of sentence structure. *American Mathematical Monthly* 65:154–170.

Lappin, Shalom. 1996. Generalized quantifiers, exception phrases, and logicality. *Journal of Semantics* 13:197–220.

Lasnik, Howard and Juan Uriagereka. 1988. *A Course in GB Syntax*. Cambridge, MA: The MIT Press.

Lasnik, Howard and Juan Uriagereka with Cedric Boecks. 2003. *A Course in Minimalist Syntax*. Blackwell.

Lassaigne, Richard and Michel de Rougement. 1993. *Logique et Fondements de L'Informatique*. Paris: Hermes.

Lewis. 1975. Adverbs of quantification. In *Formal Semantics of Natural Language*, pages 3–15. Cambridge University Press.

Lindström, P. 1966. First order predicate logics with generalized quantifiers. *Theoria* 35:186–195.

Linebarger, Maria. 1987. Negative polarity and grammatical representation. *Linguistics and Philosophy* 10:325–387.

Lønning, Jan Tore. 1985. Plurals and collectivity. In van Benthem and ter Meulen (1985), pages 1011–1059.

Mates, Benson. 1972. *Elementary Logic*. NY: Oxford University Press, 2nd edn.

Matthewson, Lisa. 2008. *Quantification: a Cross-Linguistic Persepctive*. Bingley.

McCarthy, J. 2003. Sympathy, cumulativity, and the duke-of-york gambit. In C. Féry and R. van de Vijver, eds., *The Syllable in Optimality Theory*, pages 23–76. Cambridge University Press.

McCawley, J. 1982. Parentheticals and discontinuous constituent structure. *Linguistic Inquiry* 13:99–107.

McCawley, James D. 1988. *The Syntactic Phenomena of English*, vol. 1. Chicago: The University of Chicago Press.

Miller, George A. and S. Isard. 1964. Free recall of self-embedded English sentences. *Information and Control* 7:292–303.

Milsark, Gary. 1977. Towards and explanation of certain peculiarities in the existential construction in English. *Linguistic Analysis* 3:1–30.

Moltmann, Friedericke. 1996. Resumptive quantification in exception sentences. In *Quantifiers, Deduction, and Context*, pages 139–170. Stanford, CA: CSLI Publications.

Monk, Donald J. 1976. *Mathematical Logic*. Springer-Verlag.

Montague, Richard. 1970. Universal grammar. In R. Thomason, ed., *Formal Philosophy: Selected Papers of Richard Montague*, pages 222–247. New Haven, CT: Yale University Press.

Montague, Richard. 1974. English as a formal language. In R. Thomason, ed., *Formal Philosophy: Selected Papers of Richard Montague*, pages 188–221. New Haven, CT: Yale University Press.

Moortgat, Michael. 1996. Categorial type logics. In *Handbook of Logic and Language*, pages 73–178. Kluwer Academic Publishers.

Mortimer, Michael. 1975. On languages with two variables. In *Zeitschrift fuer mathematische Logik und Grundlagen der Mathematik*, pages 135–140. Bd. 21.

Mundy, J. L. and A. Zisserman. 1992. *Geometric Invariance in Computer Vision*. MIT Press.

Nam, Seungho. 1994. Another type of negative polarity item. In Kanazawa and Piñón (1994).

Oehrle, Richard, Emmon Bach, and Deirdre Wheeler. 1988. *Categorial Grammars and Natural Language Structures*. Reidel.

Ojeda, A. 1987. Discontinuity and phrase structure grammar. In A. Manaster-Ramer, ed., *Mathematics of Language*, pages 257–277. John Benjamins.

Özkaragöz, Inci. 1986. Mono-clausal double passives in Turkish. In D. Slobin and K. Zimmer, eds., *Studies in Turkish Linguistics*, pages 77–93. John Benjamins.

Partee, Barbara. 1985. Noun phrase interpretation and type-shifting principles. In J. Groenendijk et. al, ed., *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, vol. 8 of *GRASS*, pages 115–143. Foris.

Partee, Barbara. 1994. Quantificational structures and compositionality. In Kanazawa and Piñón (1994).

Partee, Barbara, Alice ter Meulen, and Robert Wall. 1990. *Mathematical Methods in Linguistics*. Dordrecht: Kluwer.

Payne, John. 1985. Negation. In T. Shopen, ed., *Language Typology and Syntactic Description*, vol. 1, pages 197–242. Cambridge, UK: Cambridge University Press.

Pederson, E. 1993. Zero negation in south dravidian. In L. Dobrin, L. Nichols, and R. Rodriguez, eds., *Parasession on Negation*, pages 233–245. Chicago Linguistics Society.

Pelletier, Francis J. and Lenhart K. Schubert. 1989. Generically speaking. In G. Chierchia, B. H. Partee, and R. Turner, eds., *Properties, Types, and Meaning*, vol. 2. Kluwer.

Perlmutter, David and Paul Postal. 1983. Towards a universal characterization of passivization. In D. Perlmutter, ed., *Studies in Relational Grammar*, vol. 1, pages 3–30. Chicago: University of Chicago Press.

Peters, Stanley and Dag Westerståhl. 2006. *Quantifiers in Language and Logic*. Oxford.

Pratt, Ian and Nissim Francez. 2001. Temporal prepositions and temporal generalized quantifiers. *Linguistics and Philosophy* 24(2):187–222.

Pullum, G. 1976. The duke of york gambit. *Journal of Linguistics* 12:83–102.

Resnik, Philip. 1992. Left-corner parsing and psychological plausibility. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING '92)*. Nantes, France.

Reuland, Eric J. and Alice ter Meulen. 1987. *The Representation of (In)definiteness*. Cambridge MA: MIT Press.

Rooth, Mats. 1987. Noun phrase interpretation in Montague grammar, file change semantics, and situation semantics. In Gärdenfors (1987), pages 237–268.

Rothstein, Susan. 1988. Conservativity and the syntax of determiners. *Linguistics and Philosophy* 26(6):999–1021.

Russell, Bertrand and Alfred North Whitehead. 1910. *Principia Mathematica*. Cambridge University Press.

Schachter, P. 1984. *Studies in the Structure of Toba Batak*, vol. 5 of *Occasional Papers in Linguistics*. UCLA.

Smessaert, Hans. 1996. Monotonicity properties of comparative determiners. *Linguistics and Philosophy* 19(3):295–336.

Steedman, Mark and Jason Baldridge. 2007. Combinatory categorical grammar. In R. Borsley and K. Borjars, eds., *Non-Transformational Syntax*. Blackwell.

Szabolcsi, Anna. 1997. Strategies for scope taking. In A. Szabolcsi, ed., *Ways of Scope Taking*, pages 109–155. Kluwer.

Thysse, Elias. 1983. *Laws of Lanquage*. Master's thesis, Filosofisch Instituut, Rijksuniversiteit, Groningen.

Trakhtenbrot, Boris. 1950. The impossibility of an algorithm for the decision problem for finite domains (Russian). *Doklady Akademii Nauk SSSR* 70:569–572. English translation in AMS Transl. Ser. 2 vol 23, 1963:1–6.

Turing, Alan M. 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* 42:230–265. Reprinted in M. Davis (editor) 1965: *The Undecidable. Basic papers on undecidable propositions, unsolvable problems, and computable functions.* Raven Press, Hewlitt, New York.

van Benthem, Johan. 1984a. Questions about quantifiers. *Journal of Symbolic Logic* 49:443–466.

van Benthem, Johan. 1984b. Tense logic and time. *Notre Dame Journal of Formal Logic* 25(1):1–16.

van Benthem, Johan. 1986. *Essays in Logical Semantics*. Dordrecht: D. Reidel.

van Benthem, Johan and Alice ter Meulen, eds. 1985. *Generalized Quantifiers in Natural Language*. Dordrecht: Foris.

van Dalen, Dirk. 2004. *Logic and Structure*. Berlin: Springer, 4th edn.

van der Does, Jaap and Jan van Eijck, eds. 1996. *Quantifiers, Logic, and Language*. Lecture Notes 54. CSLI.

Verkuyl, Henk and Jaap van der Does. 1996. The semantics of plural noun phrases. In van der Does and van Eijck (1996), pages 337–413.

von Fintel, Kai. 1993. Exceptive constructions. *Natural Language Semantics* 1(2).

Westerståhl, Dag. 1989. Quantifiers in formal and natural languages. In D. Gabbay and F. Guentner, eds., *The Handbook of Philosophical Logic*, vol. IV, pages 1–131. Reidel.

Westerståhl, Dag. 1985. Logical constants in quantifier languages. *Linguistics and Philosophy* 8:387–413.

Weyl, H. 1952. *Symmetry*. Princeton University Press.

Winter, Yoad. 1998. *Flexible Boolean Semantics: coordination, plurality and scope in natural language*. Ph.D. thesis, Utrecht University.

Winter, Yoad. 2001. *Flexibility Principles in Boolean Semantics*. Cambridge MA: The MIT Press.

Wood, Mary McGee. 1993. *Categorial Grammars*. Linguistic Theory Guides. London: Routledge.

Zuber, Richard. 2008. A semantic constraint on binary determiners. Paper presentes to the Semantics Reseach Group meeting, Tokyo Japan.

Zwarts, Frans. 1981. Negatief polaire Uitdrukkingen. *Glot* 4:35–132.

Zwarts, Frans. 1996. Facets of negation. In J. van der Does and J. van Eijck, eds., *Quantifiers, Logic, and Language*, no. 54 in Lecture Notes. CSLI Lecture Notes.