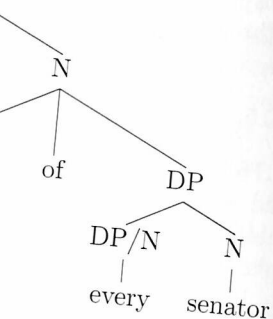


gories for the two subclasses of Dets. Note
features must be inherited by the DP so
y accordingly: *every student is laughing*
ng.

ch as *every friend of the President*, some
generate these let us add to our lexicon

ue, relative, employee

nd *either*, namely as introduced by the
category. We stipulate that MERGE ap-
P). Then we can derive:



nd of a friend of a senator, DP).

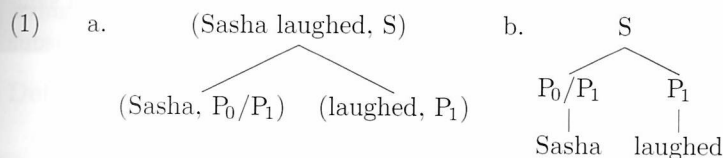
6

A Taste of Formal Language Theory

6.1 Introduction

In Chapter 5 we provided several instantiations of a grammar ENG for a fragment of English. They were given in a *bottom up* (or *inductive*) format. Here we introduce a *top down* format called *context free grammars* (CFGs), which has been widely used in linguistics and computer science. (Recall from Chapter 1: *If you can't say something two ways you can't say it*. Studying an object from different points of view improves your understanding of it—and two intertranslatable formalizations may suggest different questions and generalizations.)

Before formally defining the CFG format we provide some informal reasoning from this perspective. In the grammar ENG of Chapter 5 we say that *Sasha laughed* is a string of category S (= P_0) in $L(ENG)$. We summarized that argument with an F-A tree like (1a) or, equivalently, a standard tree like (1b).



Reading either tree from the bottom up we infer first that *Sasha* is a string of category P_0/P_1 and *laughed* a string of category P_1 , and some rule combines them to give the string in the root node, of category S.

Now reading (1b) top down, we say that an expression of category S may consist of one of category P_0/P_1 followed by one of category P_1 . And a P_0/P_1 may be *Sasha* and one of category P_1 *laughed*. We express these three steps using the context free format by:

(2) a. $S \rightarrow P_0/P_1 + P_1$

b. $P_0/P_1 \rightarrow \text{Sasha}$

c. $P_1 \rightarrow \text{laughed}$

Note that the + sign in (2) merely serves to separate two expressions (category names in the case at hand). We sometimes omit it and just leave a space to indicate the separation.

The expressions in (2) are called *productions* (or *rules*). The upper case symbols are called *non-terminal* symbols (sometimes *variables*) and the lower case ones *terminal* symbols. Mathematically the choice of terminal and non-terminal symbols is arbitrary. In modeling grammars for natural languages we choose the non-terminals to be names of grammatical categories and the terminals to be words (roughly) in the language under study. In general the productions of a CFG are of the form in (3):

- (3) $A \rightarrow w$ (A a non-terminal symbol, w a string whose coordinates may be any mix of terminal and non-terminal symbols).

The productions of a CFG, required to be finite in number, are the mechanism for deriving strings starting with non-terminal symbols. (3) says that given a string with A in it we can derive another string by replacing A with w keeping everything else the same. We illustrate how this works in a simple but characteristic case. Let G be a CFG whose only non-terminal is S and whose only terminal symbols are a and b , and whose two productions are as in (4):

- (4) 1. $S \rightarrow ab$ 2. $S \rightarrow aSb$

Here are three derivations of terminal strings (strings of terminal symbols) beginning with S :

D1. S
 ab

D2. S
 aSb
 $aabb$

D3. S
 aSb
 $aaSbb$
 $aaabbbb$

The first line in each of these derivations is the string whose only coordinate is S . All the other lines are derived by replacing an occurrence of S in the line immediately above with a string w , where $S \rightarrow w$ is a production. The last line in each derivation is a terminal string as it has no occurrences of non-terminals.

Exercise 6.1. Using the assumed CFG above

- a. exhibit a derivation of a^5b^5 .

) merely serves to separate two expressions
e at hand). We sometimes omit it and just
e separation.

are called *productions* (or *rules*). The upper
n-terminal symbols (sometimes *variables*)
terminal symbols. Mathematically the choice
al symbols is arbitrary. In modeling gram-
we choose the non-terminals to be names of
the terminals to be words (roughly) in the
neral the productions of a CFG are of the

nal symbol, w a string whose coordinates
ninal and non-terminal symbols).

required to be finite in number, are the
s starting with non-terminal symbols. (3)
A in it we can derive another string by
everything else the same. We illustrate how
characteristic case. Let G be a CFG whose
hose only terminal symbols are a and b ,
e as in (4):

$\rightarrow aSb$
terminal strings (strings of terminal sym-

S	D3.	S
aSb		aSb
aabb		aaSbb
		aaabbb

derivations is the string whose only coor-
are derived by replacing an occurrence
ove with a string w , where $S \rightarrow w$ is a
h derivation is a terminal string as it
nals.

ed CFG above

b. give a reason why aab is not derivable from S in this grammar

c. The set of terminal strings derivable from S above is $\{a^n b^n \mid n \geq 1\}$. Say informally why this is so.

6.2 CFGs: A Formal Definition

Definition 6.1. G is a *context free grammar* (CFG) iff $G = \langle N, T, S, \rightarrow \rangle$, where:

N is a non-empty set, whose elements are called *non-terminal* symbols.
 T is a set disjoint from N , whose elements are called *terminal* symbols.
 S is an element of N , called the *start* symbol, and
 $\rightarrow$ is a finite binary relation from N to $(N \cup T)^*$, that is, $\rightarrow$ is a finite subset of $N \times (N \cup T)^*$. (As usual with binary relations R we write aRb for $(a, b) \in R$; so here we write for example $S \rightarrow aSb$ rather than $\langle S, aSb \rangle \in \rightarrow$.) Elements of $\rightarrow$ are called *productions* (or *rules*). We read $A \rightarrow w$ as “ A rewrites as w ”.

This definition, given in standard format, says that to define a CFG it is necessary and sufficient to specify four things: the set of non-terminal symbols, the set of terminal symbols, the start symbol, and the productions. Often non-terminals are written in upper case, terminals in lower case. Using these conventions we can specify a CFG just by giving its productions and the start symbol, which in linguistic contexts is often S.

Definition 6.2. Given a CFG G , a *derivation* over G is a sequence $u = u_1, u_2, \dots, u_n$ of elements of $(N \cup T)^*$ such that for each i , $1 \leq i < n$, u_{i+1} results from u_i by replacing some non-terminal A in u_i with a string w , where $A \rightarrow w$ is a production of G . (Recall the definition of substitution in Chapter 2.)

Definition 6.3. Given $G = \langle N, T, S, \rightarrow \rangle$ as above,

3.1 for all $w \in (N \cup T)^*$, all $A \in N$, w is *derivable* from A iff there is a derivation $u_1, u_2, \dots, u_n$ in which $u_1 = A$ and $u_n = w$.

3.2 $L(G) = \{w \in T^* \mid w \text{ is derivable from the start symbol}\}$. $L(G)$ is called the *language generated by G*,

3.3 A set M of strings over some alphabet is *context free* (a *context free language* or *CFL*) iff there is a CFG G such that $L(G) = M$. $\square$

Using these definitions we can now formally define the CFG used in (4): $G_{SE} = \langle \{S\}, \{a, b\}, S, \rightarrow \rangle$, where $\rightarrow$ is given by: $S \rightarrow ab$ and

$S \rightarrow aSb$. (5) is a derivation of a^4b^4 . This grammar is a minimal *self embedding* grammar, where we say that G is *self embedding* if there are non-empty terminal strings u, v, w such that both w and $u + w + v$ are derivable from the start symbol. In such a case w is said to be *self embedded* in $u + w + v$. A context free stringset (= set of strings) is said to have the *self embedding* property if every CFG for that set is self embedding. $L(G_{SE})$ is a self embedding stringset. Here is a derivation of a^4b^4 which illustrates several instances of self embeddings.

(5) $S, aSb, aaSbb, aaaSbbb, aaaabbbb$

Observe that the set of strings derivable from S in the grammar in (4) is exactly the set defined by closure in Section 2.5 of Chapter 2. In the CFG literature proving that $L(G)$ is the set of strings Φ satisfying some conditions typically uses two proofs by mathematical induction: an induction on the length of derivations shows that $L(G) \subseteq \Phi$ and an induction on the length of expressions shows that $\Phi \subseteq L(G)$. These proofs are often tedious and rather than exercise the reader with them we show in an addendum how to convert CFGs to recursively defined sets as in Chapter 2 where the proof techniques there apply.

Exercise 6.2. Show that $\{a^n cb^n \mid n \geq 1\}$ is a CFL by exhibiting a grammar G such that $L(G) = \{a^n cb^n \mid n \geq 1\}$. (You are not asked to prove this equality.) Note that $\{a^n cb^n \mid n \geq 1\}$ is also a self embedding stringset.

Exercise 6.3.

- Show that $\{a^{n+1}b^n \mid n \geq 1\}$ is context free.
- Show that $\{a^n b^{n+1} \mid n \geq 1\}$ is context free.

Exercise 6.4.

- Show that any finite set is context free.
- For V a finite set show that V^* is context free.

The sets in these exercises are simple examples of CFLs (context free languages). A more difficult one is $\{w \in \{a, b\}^* \mid \text{the number of } a\text{'s in } w \text{ is the same as the number of } b\text{'s in } w\}$ (see Hopcroft et al., 2001, 181).

Notational simplification: In writing sequences, such as (5) for example, we often omit angled brackets and sometimes the commas between coordinates when no confusion will result. We often write X

$\rightarrow w|w'$ instead of writing separate productions $X \rightarrow w$ and $X \rightarrow w'$. Here we use this notation to specify a few CFGs. $\square$

To argue that a terminal string w is derivable from the start symbol of a CFG G we should, in principle, present a derivation, as in (5). Here are a few simple if artificial examples of context free stringsets. We provide CFGs for them and ask the reader to illustrate a few derivations. In all cases S is the start symbol.

- (6) $\{a^n b^m \mid n, m > 0\}$ Productions: (1) $S \rightarrow aS \mid aB$
 (2) $B \rightarrow b \mid bB$

Here is a derivation of $a^2 b^3$: $S, aS, aaS, aaB, aabbB, aabbb$

The grammar illustrated in (4) derives expressions of even length, as it derives all expressions consisting of one or more a 's followed by the same number of b 's. But many w in $\{a, b\}^*$ of even length are not generated, $aa, abab$, for example. However the set of w in $\{a, b\}^*$ of even length is context free:

Exercise 6.5. Define a CFG G such that $L(G) = \{w \in \{a, b\}^* \mid |w| \text{ even}\}$

Here is a CFG for Sentential Logic with three atomic formulas, which uses parentheses to avoid structural ambiguity.

- (7) $S \rightarrow p \mid q \mid r \mid \text{not } S \mid (S \ \& \ S) \mid (S \text{ or } S)$

Exercise 6.6. Exhibit derivations of each of the following:

- a. q b. $(p \ \& \ (q \text{ or } r))$ c. $((p \ \& \ q) \text{ or } r)$
 d. $((p \ \& \ q) \text{ or } (p \ \& \ r))$ e. $\text{not } (p \text{ or } q)$ f. $(\text{not } p \text{ or } q)$

In the CFL defined in (7) p, q , and r are called *atomic formulas*. Ordinarily in Sentential Logic (Chapter 8) the set of atomic formulas: $P_1, P_2, \dots$ is countably infinite. It might appear that we could give a CFG for this language using rules of the form $S \rightarrow P_n$ for each natural number n . But then we would have infinitely many rules, which is not allowed by the definition of CFG. Can you see a way to, in effect, build denumerably many atomic formulas using only finitely many rules? (Hint: Enrich the grammar so that P_0, P_{10}, P_{110} , etc. are formulas. These will be the atomic ones. Note that, as given, no atomic formula is a constituent of any other.)

Matching parentheses are often used in formal expressions to make constituent structure explicit in the linear string. For example in the language of elementary arithmetic $((2+3) \times 4)$ and $(2+(3 \times 4))$ are distinct expressions (and denote different numbers: 20 and 14 respectively). One check on whether formal expressions are well formed is

whether the parentheses are nested and matched correctly—for every left parenthesis there is a matching right parenthesis. Here is a CFG that generates the language of matching parentheses (Hopcroft et al., 2001, 192). We use e for the empty string (often written as an epsilon ϵ in the formal language literature).

$$(8) \quad S \rightarrow SS \mid (S) \mid ()$$

Here is a derivation of $((()()))$: $S, (S), ((S)), ((SS)), (((S))), (((()())))$.

Exercise 6.7. Exhibit derivations for each of the following:

- a. $()$ b. $((()()))$ c. $((()()))$

A *palindrome* is a sequence which reads the same backwards as forwards. In English *Anna*, *Bob*, *Mom* and *Dad* are palindromes (ignoring capitalization.) So is the noun *radar* and the adjective *level*. Longer ones are harder to find, but two are (also ignoring punctuation): *Madam I'm Adam* and, recalling Napoleon who was exiled to Elba, *Able was I ere I saw Elba*.

Exercise 6.8. Exhibit a CFG whose language is the set of palindromes in $\{a,b\}^*$.

Hint: You will need the production $S \rightarrow e$. e is a palindrome, as are a and b .

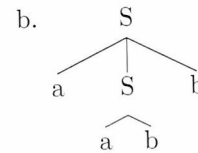
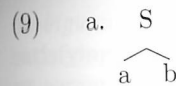
In practice linguists do not use derivations directly to show that a given string is in some $L(G)$. Rather they give a *parse tree* (or *derivation tree*).

Parse Trees and Structural Ambiguity

Definition 6.4. A parse tree for a CFG G is a finite ordered labeled tree T satisfying:

- The root node of T is labeled with the start symbol of G .
- All leaf nodes of T are labeled with terminal symbols.
- All interior (non-leaf) nodes are labeled with non-terminal symbols.
- If A labels an interior node which immediately dominates just the nodes labeled $w_1, w_2, \dots, w_k$ in order then $A \rightarrow w_1 + w_2 + \dots + w_k$ is a production of G . (We use $+$ for concatenation here.)

Here are parse trees for D1 and D2 above:



Exercise 6.9. Using the above grammar, exhibit the parse tree for $aaabbb$.

The leaf sequence of a parse tree T is called the *yield* of T . $L(G)$ is the set of yields of the parse trees for G . To *parse* a string in $L(G)$ is to provide a parse tree for it.

Fact 1. $\{ww^R \mid w \in \{a,b\}^*\}$ is context free. (w^R is w in reverse order. So if $w = \langle s_1, s_2, \dots, s_n \rangle$ then $w^R = \langle s_n, s_{n-1}, \dots, s_1 \rangle$.)

An explicit recursive definition of w^R is:

$$\text{For all } w \in V^*, w^R = \begin{cases} e & \text{if } w = e \quad (e \text{ is the empty string}) \\ b \frown y^R & \text{if } w = y \frown b \quad (\text{all } b \in V, \text{ all } y \in V^*). \end{cases}$$

Exercise 6.10.

a. Exhibit a CFG G whose language is $\{ww^R \mid w \in \{a,b\}^*\}$. Note that the set $\{ww^R \mid w \in \{a,b\}^*\}$ is just the set of palindromes of Exercise 6.8.

b. Exhibit the parse tree for $ababbaba$.

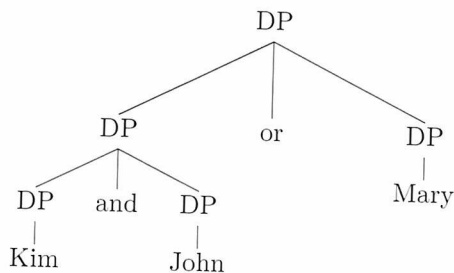
Linguists use parse trees to give a clear characterization of how expressions are built, their constituent structure. In some interesting cases they enable us to grasp quickly certain types of structural ambiguities. Consider the ambiguous grammar $G[\text{coord}]$ below:

(10) $G[\text{coord}] = \langle N, T, DP, \rightarrow \rangle$, where $N = \{DP\}$, $T = \{\text{and, or}\}$, $Sasha, Kim, John, Mary\}$, DP is the start symbol, and $\rightarrow$ is given below:

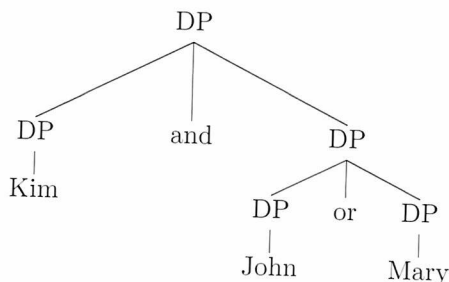
$$\begin{aligned} DP &\rightarrow DP \text{ and } DP \mid DP \text{ or } DP \\ DP &\rightarrow Sasha \mid Kim \mid John \mid Mary \end{aligned}$$

Observe that *Kim and John or Mary* is a (string of category) DP in $L(G[\text{coord}])$. In fact it has two non-isomorphic parse trees (and so two distinct derivations):

(11) a.



b.



In ordinary English these two expressions have different meanings. For example the sentence *Kim and John or Mary arrived early* may have different truth values according to how we analyze the subject. Imagine that Kim didn't arrive early and that Mary did. On the analysis in (11a) the sentence is true but on the analysis in (11b) it is false. (It doesn't matter whether John arrived early or not.)

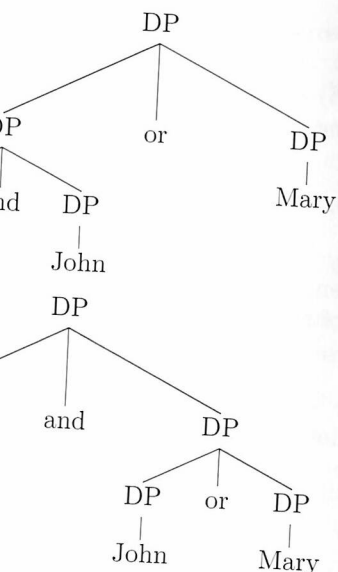
If we modified the grammar so that coordinations were flanked by matching parentheses then (11a) would appear as ((Kim and John) or Mary) and (11b) would be (Kim and (John or Mary)). While ordinary English does not use parentheses in this way the use of *both* and *either* with *and* and *or* has the same disambiguating effect: (11a) would be as in (12a), (11b) as in (12b):

(12) a. either both Kim and John or Mary

b. both Kim and either John or Mary

It is in part for representing structural ambiguities of this sort that linguists prefer to think of grammars as generating trees rather than simply strings. Two grammars may have the same *weak* generative capacity in the sense of generating the same string sets, but may differ radically in their *strong* generative capacity, namely the sets of parse trees they determine.

And the parse tree representation has another conceptual appeal. We may define an expression in $L(G)$, G context free, to be *structurally*



expressions have different meanings. For *and John or Mary arrived early* may have to do with how we analyze the subject. Imagine *and that Mary did*. On the analysis in (11a) it is false. (It arrived early or not.)

John or Mary
John or Mary

structural ambiguities of this sort that grammars as generating trees rather than strings may have the same *weak* generative capacity, namely the sets of parse

generation has another conceptual appeal. Let $L(G)$, G context free, to be *structurally*

ambiguous if it has two or more non-isomorphic parse trees. It is less satisfying to define structural ambiguity directly in terms of derivations, as sometimes distinct derivations for the same string determine the same parse tree when the expression is not judged to be structurally ambiguous. For example:

(13) Let G with start symbol S be given by:

- a. $S \rightarrow AB$
- b. $A \rightarrow a \mid aA$
- c. $B \rightarrow b \mid bB$

Here $L(G) = \{a^n b^m \mid n, m > 0\}$ for which we gave a slightly different grammar above. (14) below is one derivation of ab in this grammar:

(14) S, AB, aB, ab

Exercise 6.11.

- a. Give a second, distinct, derivation of ab .
- b. Exhibit the parse tree corresponding to each of these derivations

(Derivations are sequences and, recall, sequences are distinct iff they have different lengths or for some i , their i^{th} coordinates are distinct.)

Exercise 6.12. Exhibit a CFG G for $\{a^n b^m \mid n, m > 0\}$ in which every string generated has exactly one derivation (and hence exactly one parse tree).

Exercise 6.12 prompts the following query: Does a CFL always have an unambiguous grammar? Surprisingly perhaps the answer is no (Ginsburg, 1966, 207; Kelley, 1995, 96, and Hopcroft et al., 2001, 212–14).

Theorem 6.1 (Inherently ambiguous languages). *There exist CFLs M such that every CFG G for M is ambiguous (i.e. for every such G there is a $w \in L(G)$ such that w has at least two non-isomorphic parse trees).*

The example Kelley gives is $\{a^i b^j c^k \mid i = j \text{ or } j = k\}$. The proof that every CFG for this set is ambiguous lies well beyond what we can cover here. But we note the first step in the proof as it illustrates the utility of a basic closure property of CFGs. Namely, we must show first that the stringset above is context free. To do this, consider the stringsets in (15).

- (15) a. $\{a^i b^j c^k \mid i, j, k > 0 \text{ and } i = j\}$
- b. $\{a^i b^j c^k \mid i, j, k > 0 \text{ and } j = k\}$

Each of these sets is easily seen to be context free using the same sort of mechanism we used in (4). Here is a grammar for (15a):

$$(16) \begin{array}{l} S \rightarrow abC \mid aTbC \\ T \rightarrow ab \mid aTb \\ C \rightarrow c \mid cC \end{array}$$

Exercise 6.13. Exhibit a derivation of $a^3b^3c^2$ using the rules in (16).

Exercise 6.14. Exhibit the productions of a CFG for (15b).

So the sets in (15a) and (15b) are both context free. And we now observe:

Theorem 6.2. *The union of two context free sets is itself context free. (That is, for G and G' CFGs, $L(G) \cup L(G')$ has a context free grammar.)*

Here is an informal proof. Let G and G' be CFGs. We construct a CFG G'' for $L(G) \cup L(G')$. We may assume that the non-terminals of G are disjoint from those of G' (if not, relabel them to make them disjoint). Let S be the start symbol of G and S' that for G' . Define the productions of G'' to be the union of those of G with those of G' together with $S'' \rightarrow S \mid S'$, where S'' is the start symbol for G'' (and not a non-terminal of G or G'). Clearly $L(G'') = L(G) \cup L(G')$. $\square$

It is the second part of the proof of Theorem 6.1 that lies beyond the scope of this chapter, namely showing that *all* grammars G'' for $L(G) \cup L(G')$ above have expressions with more than one parse tree. But, following Kelley, we can point out where the problem lies. Essentially we need one subset of rules to generate $L(G)$ and another to generate $L(G')$ (or sets that differ just finitely from these. See Hopcroft et al. (2001, 212–14)). This means there will be two ways to derive expressions of the form $a^n b^n c^n$, whence the inherent ambiguity.

The existence of context free languages that are inherently ambiguous is of some mathematical and epistemological interest: it places limits on our ability to characterize the structure of expressions in some languages using certain mechanisms, those given by CFGs.

There may also be a linguistic lesson to be learned here: Linguists working on grammars of natural languages often ask *What is the structure of this expression?* But this question may not have a precise answer. If the language in question is context free and inherently ambiguous then no matter what grammar G is considered there will be expressions that have more than one structure (parse tree) in G . A sobering thought.

seen to be context free using the same sort of argument as in (4). Here is a grammar for (15a):

derivation of $a^3b^3c^2$ using the rules in (16).

the productions of a CFG for (15b).

(15b) are both context free. And we now

of two context free sets is itself context free. If G and G' are CFGs, $L(G) \cup L(G')$ has a context free

of. Let G and G' be CFGs. We construct a new CFG G'' . We may assume that the non-terminals of G and G' are disjoint (if not, relabel them to make them so). Let S'' be the start symbol of G'' . Define G'' to be the union of those of G with those of G' where S'' is the start symbol for G'' (and S is the start symbol for G and S' is the start symbol for G'). Clearly $L(G'') = L(G) \cup L(G')$. $\square$

the proof of Theorem 6.1 that lies beyond merely showing that all grammars G'' for a language L have expressions with more than one parse tree. It simply points out where the problem lies. Es- sentially, the problem is that there are two sets of rules to generate $L(G)$ and another to generate $L(G')$. These two sets of rules differ just finitely from these. See Hopcroft and Ullman (1979) for a discussion of this. It means there will be two ways to derive ex- pressions in L , whence the inherent ambiguity.

free languages that are inherently ambigu- ous. This is of great epistemological interest: it places lim- its on the structure of expressions in some languages, those given by CFGs.

istic lesson to be learned here: Linguists studying natural languages often ask *What is the struc- ture of the language?* This question may not have a precise answer. If a language is context free and inherently am- biguous, then no grammar G is considered there will be more than one structure (parse tree) in G . A

Let us return now to (14). The moral of the example is that two or more occurrences of non-terminals in a line of a derivation can be rewritten in any order without changing the parse tree. So the am- biguity is an artifact of the CFG notation but does not correspond to a pretheoretically felt structural ambiguity. To avoid such multiple derivations we often stipulate that symbol replacement in a derivation always affect the leftmost (or the rightmost) non-terminal first.

Note though that while this, quite arbitrary, stipulation eliminates certain trivial cases of multiple derivations it does not eliminate struc- tural ambiguity.

Exercise 6.15. Exhibit two leftmost derivations of *Kim and John or Mary* in $G[\text{coord}]$ given in (10). Note that the two derivations still correspond to different ways of semantically interpreting the derived expression.

Exercise 6.16. Consider the small arithmetic grammar below and exhibit two leftmost derivations of $2+3 \times 4$: $S \rightarrow S+S \mid S \times S \mid 2 \mid 3 \mid 4$

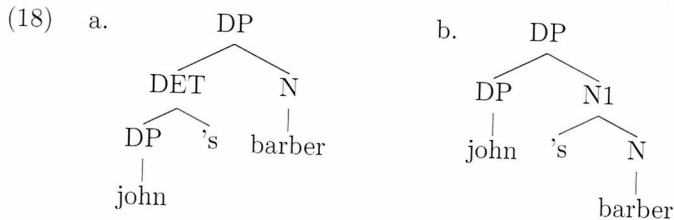
Linguists' concern with the structure of expressions recalls a fun- damental issue adumbrated in Chapter 1: If we think of a grammar of a natural language as representing its structure, and, as linguists are wont to do, we think of the structure of a language as (at least partially) reflecting the structure of mind, then we are quickly led to query the extent to which the set of expressions in a language determines the grammar. Suppose we have an oracle who can tell us for each string s of English words (roughly) whether s is a well formed expression of En- glish or not. Then, to what extent does that information enable us to determine a (generative) grammar for English (possibly an ambiguous one)? In fact, given a CFG G we can always find many other G' that are weakly equivalent to G in the sense that they generate the same set of strings. Some are trivial variants of each other to be sure. Given G we could make up a trivial variant by adding some non-terminals to the grammar that were never mentioned in any production. So this would be technically a different grammar but one that generated the same expressions. Without trying to define the class of trivial variants of grammars we can nonetheless notice that it is not unreasonable to find structurally distinct grammars that generate the same expressions. Here is a candidate simply to give the reader some idea of substantive alternative analyses that arise in linguistic analysis.

(17) Let G_1 and G_2 be as below, with, in both cases, DP the start symbol and 's, john, mary, doctor, barber the terminal symbols:

G_1 : $DP \rightarrow DET + N$
 $DET \rightarrow DP + 's$
 $DP \rightarrow \text{john} \mid \text{mary}$
 $N \rightarrow \text{doctor} \mid \text{barber}$

G_2 : $DP \rightarrow DP + N1$
 $N1 \rightarrow 's + N$
 $DP \rightarrow \text{john} \mid \text{mary}$
 $N \rightarrow \text{doctor} \mid \text{barber}$

(18a) is a parse tree for *john's barber* using G_1 , and (18b) using G_2 .



In fact linguists have provided arguments in favor of both of these analyses. So we might decide it is a case of structural ambiguity, like the coordination example above but not implying a semantic ambiguity, or we might opt for one or the other based on evidence that goes beyond mere “descriptive adequacy”—generating the expressions in question. For example, perhaps the different analyses generalize differently to new types of cases; perhaps we have evidence from semantic analysis which prefers one to the other; perhaps there is evidence from child language acquisition favoring one over the other. Sometimes competing analyses differ in “elegance”, though that does not seem to be the case with G_1 and G_2 above.

Exercise 6.17. Provide the parse trees for *john's barber's doctor* using G_1 and G_2 .

Reference A recent and very readable introduction to the formal study of context-free grammars as well as of finite state grammars discussed in the next chapter is Sipser (2012).

6.3 How Well Do CFGs Model Natural Languages?

This is really two questions: a sufficiency one and a necessity one.

Sufficiency Can enlightening grammars for all natural languages be given in a context free format? If not we need to enrich their generative mechanisms to increase the class of languages they define.

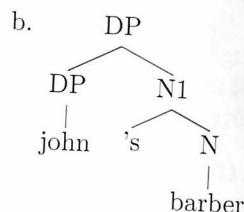
Necessity Are all languages generated by CFGs possible natural languages? If not we need to restrict their generative mechanisms to rule out certain languages.

There is modest agreement in the literature that CFGs are not sufficient, but that we do not have to enrich them much to obtain

+ N
- 's
mary
barber

G_2 : $DP \rightarrow DP + N1$
 $N1 \rightarrow 's + N$
 $DP \rightarrow \text{john} \mid \text{mary}$
 $N \rightarrow \text{doctor} \mid \text{barber}$

n's barber using G_1 , and (18b) using G_2 .



ded arguments in favor of both of these
e it is a case of structural ambiguity, like
ove but not implying a semantic ambiguity,
other based on evidence that goes beyond
—generating the expressions in question.
erent analyses generalize differently to new
ve evidence from semantic analysis which
aps there is evidence from child language
he other. Sometimes competing analyses
at does not seem to be the case with G_1

arse trees for *john's barber's doctor* using

ry readable introduction to the formal
s as well as of finite state grammars dis-
ipser (2012).

s Model Natural Languages?

sufficiency one and a necessity one.
taining grammars for all natural lan-
e format? If not we need to enrich their
ase the class of languages they define.
ges generated by CFGs possible natural
restrict their generative mechanisms to

in the literature that CFGs are not
have to enrich them much to obtain

approximate models of NLs (natural languages). The necessity question is less often asked, and it does seem *prima facie* that many CFLs are very “unnatural”.

We begin with support for the sufficiency hypothesis.

6.3.1 Properties of Natural Languages that are Expressible by CFGs

First, grammatical categories and the attendant substitution property are built into CFGs. By ‘substitution’ here we just mean that whenever some σ of category C is a constituent of an expression τ then σ can be replaced in τ by another expression σ' of the same category as σ preserving grammaticality. This enables us to represent many expressions by giving their grammatical form. For example, a Sentence consists of a DP followed by a P1. Given one S of this form we form many more (usually infinitely many more) just by replacing the DP with another DP or the P1 with another P1. Substitution applies crucially to constituents of expressions, not to strings. We have seen that *John laughed* is a sentence of English but that string is not a constituent of *The man who offended John laughed*, and that string cannot be replaced by a random sentence preserving grammaticality: **The man who offended now it is raining*.

Second, CFGs naturally represent constituent structure and structural ambiguities. Even operations which derive expressions from expressions by reordering parts or deleting parts are typically sensitive to constituent structure: what they move or delete are constituents.

Third, basic recursion is naturally represented in CFGs. It is unproblematic to generate *a teacher who criticized a teacher who criticized a teacher*. . . . So CFGs provide a finite mechanism (the productions) for deriving infinite sets of strings.

Fourth, our definition of a CFG G requires that the set of productions be finite. This is much more than a simplifying assumption, or an assumption of convenience. Were we to allow infinite sets of productions then any set whatever could be, trivially, generated by such a grammar: Let M be any set and G the grammar whose productions are $S \rightarrow m$ for each $m \in M$. Then $L(G) = M$, but we have said nothing about the structure of the elements of M .

With the finiteness restriction as we have it there are many stringsets that are provably not generable from any CFG:

Fact 2. Each of the sets below is not context free (has no CFG)

- | | |
|--|---|
| a. $\{a^n b^n c^n \mid n > 0\}$ | b. $\{a^n b^n a^n \mid n > 0\}$ |
| c. $\{a^m b^n a^m b^n \mid m, n > 0\}$ | d. $\{a^i b^j c^k \mid i \leq j \leq k\}$ |

For proof that the set in (a) above is not context-free see Ginsburg (1966, 84); for that in (c) see Hopcroft and Ullman (1979, 128).

Exercise 6.18. Show $\{a^n a^n a^n \mid n > 0\}$ is context free. This set is just $\{a^{3n} \mid n > 0\}$

The examples in **Fact 2** above show that CFGs are limited in their ability to impose conditions on triples, quadruples, and in general n -tuples of expressions for $n > 2$. Note that 2 itself is fine: the pattern $a^n w b^n$ is a fundamental pattern in CFLs and shows up in common agreement patterns. Shortly we show how such patterns may be represented in a context free system and we also consider some apparent counterexamples in natural languages.

Fifth, to claim that natural language grammars are context free is a very strong claim: many sets, per **Fact 2** above, are not CFLs. In general linguists would like the class of grammars they can draw from in modeling a natural language to be structurally very limited. This would help explain why children learn languages rather quickly with only limited exposure to, often, imperfect data (interruptions, noisy channels, memory lapses, etc.). The idea is that learning a language requires learning a grammar for it, and the task of the learner is simplified to the extent there are few choices of possible grammars. This idea receives support from one mathematical result:

Theorem 6.3. *Given an arbitrary CFG G there is a mechanical procedure which tells us for each terminal string w whether $w \in L(G)$ or not. (So it tells us it is in $L(G)$ if it is and it isn't if it isn't.) In short, given a CFG G , membership in $L(G)$ is decidable (Synonyms: solvable, computable, recursive).*

Many sets of expressions we work with as scholars are not decidable. For example the set of sentences provable from the (first order) axioms of arithmetic is not a decidable set. There is no procedure which looks at an arbitrary formula in arithmetic and says **yes** if that formula is a theorem and **no** if it isn't. On the other hand programming languages are decidable. Your computer has to be able to tell if your input text is a well formed program or not.

A few other linguistic questions about CFGs are decidable. In particular whether a language generated by an arbitrary CFG is empty or not, and whether it is infinite or not. But many other natural questions are not decidable—there is no mechanical procedure that answers the question. For example, it is not decidable whether $L(G) = L(G')$, given two arbitrary CFGs G and G' . It is similarly undecidable whether $L(G) \subseteq L(G')$ and whether $L(G) \cap L(G') = \emptyset$.

(a) above is not context-free see Ginsburg
see Hopcroft and Ullman (1979, 128).

$\{a^n | n > 0\}$ is context free. This set is just

above show that CFGs are limited in their
on triples, quadruples, and in general n -
> 2. Note that 2 itself is fine: the pattern
pattern in CFLs and shows up in common
we show how such patterns may be rep-
system and we also consider some apparent
languages.

ral language grammars are context free is
ets, per **Fact 2** above, are not CFLs. In
he class of grammars they can draw from
age to be structurally very limited. This
lren learn languages rather quickly with
en, imperfect data (interruptions, noisy
). The idea is that learning a language
for it, and the task of the learner is sim-
few choices of possible grammars. This
e mathematical result:

bitrary CFG G there is a mechanical pro-
terminal string w whether $w \in L(G)$ or
) if it is and it isn't if it isn't.) In short,
 $L(G)$ is decidable (Synonyms: solvable,

work with as scholars are not decidable.
s provable from the (first order) axioms
set. There is no procedure which looks
metic and says **yes** if that formula is a
he other hand programming languages
as to be able to tell if your input text

ons about CFGs are decidable. In par-
rated by an arbitrary CFG is empty or
r not. But many other natural ques-
no mechanical procedure that answers
not decidable whether $L(G) = L(G')$,
 G' . It is similarly undecidable whether
 $\cap L(G') = \emptyset$.

Sixth, returning now to some less mathematical merits of CFGs, we see that CFGs can express a variety of cross constituent dependencies that are prominent in many natural languages. Here are two types of examples. First, *agreement* phenomena, such as subject-predicate agreement in number:

- (19) a. John daydreams. Every poet daydreams.
b. John and Bill daydream. All poets daydream.
c. *John daydream. *Every poet daydream.
d. *John and Bill daydreams. *All poets daydreams

So a singular subject, like *John* or *every poet* requires a singular predicate, e.g. *daydreams* not *daydream*. Similarly plural subjects require plural predicates. In English we also find a kind of agreement (or selection) between DETs and Ns:

- (20) a. all poets, most poets *all poet, *most poet
b. every poet, a poet *every poets, *a poets

And in these cases the DPs formed inherit the number feature, as we see in the subject-predicate agreement in (19). Below we illustrate how agreement of the sorts noted can be incorporated in a CFG.

The idea is to enrich the system of grammatical categories, annotating them with features like +pl and -pl. So a DP[+pl] is a plural DP, a DP[-pl] is a singular one. Then we enrich our productions with subcategory rules, (21), and agreement rules, (22):

- (21) DP[-pl] $\rightarrow$ John | Mary
N[-pl] $\rightarrow$ doctor | poet
N[+pl] $\rightarrow$ doctors | poets
P1[-pl] $\rightarrow$ daydreams | drinks
P1[+pl] $\rightarrow$ daydream | drink
DP[+pl]/N[+pl] $\rightarrow$ all | most
DP[-pl]/N[-pl] $\rightarrow$ every | a

- (22) S $\rightarrow$ DP[α pl] + P1[α pl] all $\alpha \in \{+, -\}$.
DP[α pl] $\rightarrow$ DP[α pl]/N[α pl] + N[α pl] all $\alpha \in \{+, -\}$.
DP[+pl] $\rightarrow$ DP[α pl] and DP[β pl] all $\alpha, \beta \in \{+, -\}$.

Exercise 6.19.

- a. Exhibit the parse tree for *John and most poets daydream*
- b. Say why *a poet daydream* is not derived from S in this grammar.
- c. Why is *each poets* not derived in this grammar?

Thus in an S consisting of a DP and a P1, both the DP and the P1 are +pl or both are -pl. Similarly when DETs combine with Ns to form DPs the DET and N agree in plurality and the DP inherits that specification.

Agreement systems (see Corbett, 1991, 2000) are much more complex than this in many languages, but still the set of agreement features in any language is always finite so enriching rewrite rules as above is in principle always possible. We are not saying that our brute force approach to agreement is the best way to represent it, only that basic agreement phenomena lie within the purview of CFGs. If one could show for example that there were agreement patterns in natural languages that could not be represented in CFGs that would be a lethal blow to claiming that natural languages are context free.

We should note that it is not uncommon across languages to find more than two agreement morphemes of the same shape, suggesting perhaps that Ns can impose identity conditions on n -tuples of expressions for $n > 2$. Consider (23) from Swahili (Bantu; E. Africa. See Corbett, 1991, 43):

- (23) *kikapu kikubwa kimoja kilianguka*
 basket large one fell
 One large basket fell

Nouns in Bantu are commonly divided into 10 to 20 noun classes, marked by a prefix such as *ki* in (23). Adjectives, like “large” that modify the noun carry a copy of that prefix. The whole NP “large basket” is then in the *ki*-class of NPs and Determiners like “one” which combine with them and agree in class marking pass it along to the expression they derive. Finally when such DPs combine with predicates they trigger an agreement which is also expressed with the noun class marker occurring as a prefix on the verb, as in (23). Lest the reader think the example “exotic” we note that rich (though not identical) agreement patterns occur in Semitic (Hebrew, Arabic), Romance (French, Italian), Slavic (Russian, Polish), and many other groups. Here are a few examples:

ree for *John and most poets daydream*

dream is not derived from S in this grammar.

ot derived in this grammar?

g of a DP and a P1, both the DP and the
l. Similarly when DETs combine with Ns to
agree in plurality and the DP inherits that

Corbett, 1991, 2000) are much more com-
guages, but still the set of agreement features
finite so enriching rewrite rules as above is
le. We are not saying that our brute force
the best way to represent it, only that basic
within the purview of CFGs. If one could
re were agreement patterns in natural lan-
represented in CFGs that would be a lethal
ral languages are context free.

is not uncommon across languages to find
morphemes of the same shape, suggesting
se identity conditions on *n*-tuples of expres-
(23) from Swahili (Bantu; E. Africa. See

oja *kilianguka*
fell

only divided into 10 to 20 noun classes.
i in (23). Adjectives, like “large” that mod-
that prefix. The whole NP “large basket” is
and Determiners like “one” which combine
ss marking pass it along to the expression
ch DPs combine with predicates they trig-
also expressed with the noun class marker
verb, as in (23). Lest the reader think the
at rich (though not identical) agreement
Hebrew, Arabic), Romance (French, Ital-
, and many other groups. Here are a few

- (24) a. *Ha-yeladim ha-ktanim yeshenim*
The-child.m.pl the-small.m.pl sleep.m.pl.
Hebrew

The little boys are sleeping

- b. *Ha-yeladot ha-ktanot yeshenot*
the-child.fem.pl the-small.fem.pl sleep.fem.pl
The little girls are sleeping

In these cases the adjective “small” and the gender marked noun “child” agree in definiteness, gender and number. Linguists naturally generate this in several binary steps: (1) agreement (definiteness, gender, number) between the adjective and the NP, and (2) agreement between the DP subject (which inherits the gender and number features of its parts) with the predicate. Note finally the marked agreement in the French examples below:

- (25) a. *Les jeunes garçons sont heureux*
the young boys are happy
b. *Les jeunes filles sont heureuses*
the young girls be.pl happy.fem.pl

In both cases the definite article *les* ‘the’ is plural (singular masculine would be *le*, singular feminine *la*.) The *-s* on “young” and “boy/girl” marks plural (but does not change the pronunciation of the singular), *sont* is the 3rd person plural present tense form of “be”, and *heureux* is the masculine plural form of “happy”, which differs in pronunciation from the feminine plural form *heureuses* in (25b).

And in fairness let us note that while agreement phenomena are widespread in NLS, there are also many languages which present no agreement between subjects and verbs or between adjectives or determiners and nouns: Mandarin Chinese, Vietnamese, and Malagasy (Austronesian, Madagascar) are examples.

A second major type of cross constituent dependency, *filler-gap* dependencies, is illustrated by relative clause formation in Chapter 5. Copying our MERGE rules in a context free format is quite straightforward. And now we use [DP] as a feature: for C a category, C[DP] is read as “C with an DP gap”, and more generally, for C’ any non-terminal, C[C’] is a “C with a C’ gap”. (Below we partition the rule set for ease of comprehension.)

- (26) $DP/N \rightarrow \text{every} \mid a$
 $N \rightarrow \text{student} \mid \text{poet} \mid N + N/N$
 $DP \rightarrow \text{Sasha} \mid \text{Kim} \mid DP/N + N$
 $S \rightarrow DP + P_1$
 $P_1 \rightarrow \text{laughed} \mid \text{cried} \mid P_2 + DP$
 $P_2 \rightarrow \text{praised} \mid \text{criticized}$
- $DP[DP] \rightarrow t$
 $S[DP] \rightarrow DP[DP] + P_1 \mid DP + P_1[DP]$
 $P_1[DP] \rightarrow P_2 + DP[DP] \mid P_2[DP] + DP$
- $(N/N)/(S[DP]) \rightarrow \text{who} \mid \text{that}$
 $N/N \rightarrow (N/N)/(S[DP]) + S[DP]$

$C \rightarrow C$ and C , all $C \in \{DP, S, P_1, P_2, S[DP], P_2[DP]\}$

Exercise 6.20. Using the grammar in (26), exhibit the parse trees for:

- a. every student who Sasha praised t
- b. a poet who t laughed and praised Sasha

Exercise 6.21. Writing DP_{wh} to abbreviate $(N/N)/(S[DP])$ add the following productions to (26): $DP_{wh} \rightarrow DP_{wh}/N + N$ and $DP_{wh}/N \rightarrow \text{whose}$. Exhibit a parse tree for *a student whose teacher Kim praised t*.

6.3.2 Sufficiency: Are Natural Languages Adequately Modeled by CFGs?

By *adequately modeled* we understand that we are concerned with strong generative capacity—we must not only generate the right stringset, we must generate each string with a correct constituent analysis. This is needed both to provide the right input to complex processes—movement and deletion (and perhaps copying, see below) that build complex expressions from simpler ones. We also need the right constituent structure to provide the right input to processes of semantic and phonological interpretation.

That said, the CFG Hypothesis—that NLs are adequately modeled by CFGs—is useful even if we find NL properties not naturally expressible by CFGs. The reason is that, as we have seen, CFGs do capture several basic properties of NLs, so a given proposal for a grammar of a NL can be informatively described in terms of the ways it differs from a CFG. We turn now to some specific issues.

$N + N/N$
 $DP/N + N$

$P_2 + DP$
 ed

$[DP + P_1[DP]$
 $P_2[DP] + DP$

that
 $S[DP]$

$\{DP, S, P_1, P_2, S[DP], P_2[DP]\}$

grammar in (26), exhibit the parse trees for:

a praised t

nd praised Sasha

wh to abbreviate $(N/N)/(S[DP])$ add the
 $DP_{wh} \rightarrow DP_{wh}/N + N$ and $DP_{wh}/N \rightarrow$
 or a student whose teacher Kim praised t.

Natural Languages Adequately

understand that we are concerned with
 we must not only generate the right
 each string with a correct constituent
 a to provide the right input to complex
 etion (and perhaps copying, see below)
 s from simpler ones. We also need the
 provide the right input to processes of
 erpretation.

thesis—that NLs are adequately modeled
 ind NL properties not naturally express-
 hat, as we have seen, CFGs do capture
 so a given proposal for a grammar of a
 bed in terms of the ways it differs from
 specific issues.

Self Embedding (Center Embedding)

Self embedding (center embedding) was the first formal property of languages cited in support of the utility of CFG grammars for English (Chomsky, 1956, 1959). It is easily expressed, as per G_{SE} given for (4). Certain simpler classes of grammars, called *finite state* ones which define a class of languages called *regular* languages (Chapter 7) are easily shown not to accommodate self embedding. But English appears to instantiate it:

(27) $[_S \text{ That } [_S \text{ Ed left}] \text{ surprised Mary}]$

Here *Ed left* is a sentence properly embedded in the S (27). But one robin doesn't make a spring. Treating *surprised Mary* as a kind of P_1 , let's call it a P_1^* , we might posit a rule for a CFG for English such as (28).

(28) $S \rightarrow \text{that} + S + P_1^*$

We could express this in a more sophisticated way, making *that+S* a kind of sentential DP and P_1^* the kind of P_1 that combines with such DP s to make S s. But (28) suffices here, as all we want is any rule set that in one or several steps properly embeds S s within S s. And such CFGs will derive S s like (27). The problem is that they also derive (29a,b) as S s.

- (29) a. That that Ed left surprised Mary annoyed Dan
 b. That that that Ed left surprised Mary annoyed Dan pleased Sue

But the expressions in (29a,b) are not acceptable English. Suppose we decide that in structures of the form $[\text{that } [_S X] \text{ surprised Mary}]$ X cannot itself have any self embeddings. Then we do not need a self embedding rule (for these cases) and we can “fake it” by invoking a new category, say S^* , whose expansion rules exhibit no self embedding of any category. Then (28) becomes $S \rightarrow \text{that} + S^* + P_1^*$, and we capture the one level embedding without overgenerating.

Note further, the meanings of (29a,b) are naturally expressible by syntactic variants which extrapose the sentential subject:

- (30) a. It annoyed Dan that it surprised Mary that Ed left
 b. It pleased Sue that it annoyed Dan that it surprised Mary that Ed left

But these expressions do not exhibit self embedding! No sentential sub-constituent has any material to its right, so it is not properly embedded. All the recursion in these cases is right peripheral and this sort of recursion can be handled with the more restricted finite state grammars (Chapter 7).

A different instance of this phenomenon is center embedding in relative clauses. Consider the frame in (31a) when we replace X with various DPs:

- (31) a. [DP the cat that [DP X] bit t]
 b. [DP the cat that [DP my dog] bit t]
 c. [DP the cat that [DP the dog I chased t] bit t]
 d. [DP the cat that [DP the dog that [DP the man I saw t] chased t] bit t]

Here (31c) results from replacing X in (31a) with *the dog I chased*. It is at the limit of comprehensibility. One more replacement, as in (31d), crosses that line. And again right peripheral recursion on DP is more comprehensible:

- (32) a. the dog that t bit X
 b. the dog that t bit the child who t called me
 c. the dog that t bit the child who t called the man who t was smiling

Patterns such as these make one hesitate to say that self embedding is a basic property of English. It seems not to iterate easily. However, maybe that is just an accident of the specific examples we chose. Perhaps there are others that are better. And indeed even the problematic (29a,b) can be improved (somewhat) if we use the sentential subjects as complements of nouns:

- (33) a. The fact that Ed left surprised Mary
 b. The claim that the fact that Ed left surprised Mary was denied
 c. The oft-made claim that the belief that the Earth is flat was widely discussed in the Middle Ages has recently been disputed

Here (33b) is much more comprehensible than (29a), and (33c) has the same level of embedding as (33b) but is merely a little long, not incomprehensible. Further, it is possible that self embedding is more tolerated

not exhibit self embedding! No sentential sub-
al to its right, so it is not properly embedded.
e cases is right peripheral and this sort of
with the more restricted finite state grammars

f this phenomenon is center embedding in
the frame in (31a) when we replace X with

- [DP X] bit t]
- [DP my dog] bit t]
- [DP the dog I chased t] bit t]
- [DP the dog that [DP the man I saw t]

acing X in (31a) with *the dog I chased*. It is
bility. One more replacement, as in (31d),
right peripheral recursion on DP is more

X
the child who t called me
the child who t called the man who t was

one hesitate to say that self embedding is
It seems not to iterate easily. However,
t of the specific examples we chose. Per-
better. And indeed even the problematic
ewhat) if we use the sentential subjects

ft surprised Mary
fact that Ed left surprised Mary was

that the belief that the Earth is flat
l in the Middle Ages has recently been

ehensible than (29a), and (33c) has the
) but is merely a little long, not incom-
le that self embedding is more tolerated

in other languages. Recall example (26) in Chapter 1 from Nepali. It
exhibits two levels of proper self-embedding and is, impressionistically,
less problematic than (29a) above in English. So we shall continue to
require that natural language grammars allow self embedding.

Copying vs Mirror Image

We begin this section with a thought experiment. We're going to type
a sequence of five letters and numbers and ask you to read it once, close
your eyes and repeat it. Here it is: 7, Q, 5, B, N. Now close and repeat.
Likely you were successful. In general human beings seem competent
to repeat phrases that lie within their immediate memory span. But
now, glance at it again: then repeat it backwards. If you are like us you
probably repeat in your mind the whole thing and say the last letter,
then you repeat again the first four items and say the last one of that,
and so on until you are done. In short, humans seem good at repetition
but bad at making palindromes, repeating backwards.

To find longish palindromes—*Able was I ere I saw Elba*—we sought
examples from the literature as we couldn't make up any interesting
ones off the top of our head. But the set of palindromes over a finite al-
phabet is easy to generate using a CFG. Phonological copying however
is not. The top down format + context freeness (free substitutivity of
items with the same category) of CFGs blocks us from building new
strings from already derived strings. But we claim that NLs make pro-
ductive use of copying—copying of phonological strings which embody
a non-trivial amount of morphological and syntactic structure. Here we
review a few examples, as the copying processes are not well attested
in English but are in many other languages so they are something that
a specification of “possible human language” must take into account.

First, as is well known, certain kinds of word and phrase repetition—
reduplication—are widely used in natural languages, often to indicate
pluralization, intensification, repetition, or indefinitization. For ex-
ample in Malay from *akar* ‘(a) root’ $\Rightarrow$ *akar-akar* ‘many kinds of
roots’; *bunga* ‘flower(s)’ $\Rightarrow$ *bunga-bunga* ‘all sorts of flowers’, *perempuan*
‘woman’ $\Rightarrow$ *perempuan-perempuan* ‘women’. Expressions in virtually
all categories except prepositions appear to undergo reduplication, even
various indefinite pronouns: *siapa* ‘who’ $\Rightarrow$ *siapa-siapa* ‘whoever, any-
one; *apa* ‘what’ $\Rightarrow$ *apa-apa* ‘whatever, whichever, any’. Reduplication
is sufficiently common that it is often indicated in texts with a super-
script on the word: *baju*² = *baju-baju* ‘clothes’.

Of course if only lexical items, finite in number, could be redu-
plicated this would not prevent the language from being context free.
We could simply add each reduplicated item to the lexicon as a sepa-

rate word, nearly doubling its size but still remaining finite. But this approach fails to be linguistically revealing. It ignores a regular way of deriving words whose meaning is understood as a function of what they are derived from. *Bunga* 'flower' and *bunga-bunga* 'all sorts of flowers' are semantically related, unlike two independent lexical items, say *bunga* and *perempuan* 'woman'. Merely listing reduplicated forms as separate lexical entries does not enable us to predict that when a new noun is borrowed speakers can use its reduplication even though that may not even exist in the donor language.

Reduplication as in Malay is widespread in Austronesian (Tagalog, Malagasy) and also prominent in Mandarin Chinese. We note that verbal reduplication in Mandarin and in Malagasy often has an attenuating effect rather than an intensifying one. From Mandarin (Li and Thompson, 1981):

- (34) jiāo 'teach' $\Rightarrow$ jiāo-jiao 'teach a little' (second copy neutral tone)
 bèi 'recite' $\Rightarrow$ bèi-bei 'recite a little'
 yánjiū 'research' $\Rightarrow$ yánjiū-yánjiū 'research a little' (not reduce tone)

- (35) a. *mitsiky izy* b. *mitsikitsiky izy*
 smile 3s smile a bit 3s
 "He is smiling" "He is smiling a bit"

Classical reduplication is largely limited (but see Blust, 2001) to single words or even roots. In general it does not iterate: from *mitsikitsiky* 'smiles a bit' we cannot form *mitsikitsikitsikitsiky* 'smiles just a tiny tiny bit'. Productive reduplication would provably take us outside the class of CFLs. But even the more limited kind that is well attested in NL is unnatural in a CFL. We can "fake" it by entering each reduplicated verb and adjective as its own entry in the Lexicon, but that misses the generalization that we build novel expressions by copying, and we semantically interpret the result as a function of what was copied. For example, the Malagasy borrowed *latabatra* 'table' from French, using it to mean 'table', and when hearing its reduplicated form *latabatabatra* they do not wonder if it means "blind unicorn", they infer immediately that it means "something that is sort of a table, say something that is functioning as a table but wasn't really made as a table". So the Reduplication function in Malagasy is one that has a finite domain, but still a large one—numbering in the thousands, as essentially all content words—Nouns, Verbs, Adjectives and even some quantifiers reduplicate. Speakers do not learn the meanings of the reduplicated forms independently of those of the forms they are derived from. Our inter-

ling its size but still remaining finite. But this
inguistically revealing. It ignores a regular way
e meaning is understood as a function of what
Bunga 'flower' and bunga-bunga 'all sorts of
y related, unlike two independent lexical items,
an 'woman'. Merely listing reduplicated forms
ies does not enable us to predict that when a
speakers can use its reduplication even though
in the donor language.

Malay is widespread in Austronesian (Tagalog,
ominant in Mandarin Chinese. We note that
Mandarin and in Malagasy often has an atten-
an intensifying one. From Mandarin (Li and

-jiao 'teach a little' (second copy neutral tone)
pei 'recite a little'
> yánjiū-yánjiū 'research a little' (not reduce

- b. *mitsikitsiky izy*
smile a bit 3s
g" "He is smiling a bit"

largely limited (but see Blust, 2001) to single
general it does not iterate: from *mitsikitsiky*
rm *mitsikitsikitsikitsiky* 'smiles just a tiny tiny
ation would provably take us outside the class
more limited kind that is well attested in NL
e can "fake" it by entering each reduplicated
own entry in the Lexicon, but that misses
e build novel expressions by copying, and we
result as a function of what was copied. For
orrowed *latabatra* 'table' from French, using it
n hearing its reduplicated form *latabatabatra*
means "blind unicorn", they infer immedi-
ething that is sort of a table, say something
ole but wasn't really made as a table". So the
Malagasy is one that has a finite domain, but
ng in the thousands, as essentially all content
jectives and even some quantifiers redupli-
urn the meanings of the reduplicated forms
the forms they are derived from. Our inter-

est here however goes well beyond classical reduplication. We consider
first some cases where a certain amount of complex syntactic structure
is reduplicated. One well known case is A-not-A question formation in
Mandarin Chinese (Li and Thompson, 1981, Radzinski, 1990).

- (36) a. *ta zai jia* b. *ta zai jia bu zai jia?*
3s at home 3s at home not at home
S/he is at home Is s/he at home?
c. *ni xihuan ta de chenshan bu xihuan ta de*
you like 3s gen shirt not like 3s gen
chenshan?
shirt
Do you like his/her shirt?

Both Bambara (Niger-Congo; W. Africa) and Malagasy (Austronesian;
Madagascar) have a construction which copies common noun phrases
yielding a whatever reading. From Bambara (Culy, 1985) we see the
repetition in (37a), and in (37b) that non-repetition is ungrammatical.
(38a,b) show that the repeated items are productively formed. (We
have added the hyphens to mark morpheme boundaries.)

- (37) a. *wulu o wulu* b. **wulu o malo*
dog o dog *dog o rice
whatever dog
(38) a. *wulu-nyinina o wulu-nyinina*
dog-searcher o dog-searcher
whatever dog searcher
b. *wulu-nyinina-nyinina o wulu-nyinina-nyinina*
dog-searcher-searcher o dog-searcher-searcher
whoever searches for dog searchers

In Malagasy (see Rajemisa-Raolison, 1971, 73; Dez, 1990; Paul, 2005)
copying builds on interrogatives, the simplest cases reminiscent of
Malay above:

- (39) a. *na iza na iza* b. *na inona na inona*
or who or who or what or what
whoever/anyone whatever/anything

(40) (Dez, 1990)

na fanafody inona na fanafody inona no omena azy,
 or medicine what or medicine what foc given him
 whatever medicine is given to him, ...

In both Malagasy and Bambara these copy expressions are subject to performance limitations: the acceptability of copying decreases with the length of the expression.

Culy claims that the Bambara construction can be iterated and presents a mathematically correct argument that Bambara with this unbounded iteration is not a CFL. In the Malagasy case, where we have been able to consult with several Malagasy linguists¹ we have clear judgments that the items governed by *na* in (39) and (40) are limited to lexical items and perhaps some fixed expressions. So again the structures do not iterate but still use thousands of interrogative phrases with lexical heads.

We note further that Malagasy does present a richer copy paradigm with expressions of the form: Predicate + Prt + Predicate, where we have two choices of Particle: *dia* indicating intensification and *fe* indicating attenuation (again thanks to joint work with Ralalaoherivony and Razanamompionona).

- (41) a. *nianatra dia nianatra Rabe*
 past.act.study dia past.act.study Rabe
 Rabe really studied
 b. *nianatra fe nianatra Rabe*
 Rabe just sort of studied

This copying admits some syntactic complexity aside from that in the bound morphology above. Thus transitive verb + indefinite objects work easily:

- (42) *mamboatra trano dia/fe mamboatra trano Rabe*
 repair house dia/fe repair house Rabe
 Rabe is really / sort of repairing houses

However richer predicates with PP modifiers seem not to copy:

- (43) a. *Mamy dia mamy toy ny tantely ity paiso ity*
 sweet dia sweet like det honey this peach this
 This peach is sweet like honey

¹Our Malagasy judgments here and further below rely on the work of the Malagasy linguists Baholy Ralalaoherivony and Ravaka Razanamompionona

na fanafody inona no omena azy,
or medicine what foc given him
given to him, ...

Bambara these copy expressions are subject to
the acceptability of copying decreases with
Bambara construction can be iterated and
correct argument that Bambara with this
a CFL. In the Malagasy case, where we
with several Malagasy linguists¹ we have
ms governed by *na* in (39) and (40) are
perhaps some fixed expressions. So again
but still use thousands of interrogative

Malagasy does present a richer copy paradigm
Predicate + Prt + Predicate, where we
dia indicating intensification and *fe* in-
marks to joint work with Ralalaoherivony

nianatra Rabe
past.act.study Rabe

a Rabe
studied

tactic complexity aside from that in the
ous transitive verb + indefinite objects

e mamboatra trano Rabe
e repair house Rabe
pairing houses

PP modifiers seem not to copy:

toy ny tantely ity paiso ity
like det honey this peach this
ke honey

and further below rely on the work of the Mala-
and Ravaka Razanamompionona

- b. **Mamy toy ny tantely dia mamy toy ny tantely*
sweet like det honey dia sweet like det honet
ity paiso ity
this peach this

A third structure type that not uncommonly uses overt phonological copying is the formation of reciprocal constructions (each other; one another). We find this in Russian (*drug drug*), Dyirbal (Pama-Nyungan; Australia) and with possessors (each other's wives) in Abkhaz (NW Caucasian; Hewitt, 1979).

Below is a striking example from Amele (Trans-New Guinea; New Guinea; Roberts, 1987). The predicate in (44b) is a complex serial verb construction in which different-subject (ds) marking is present on both parts, as is the local (singular) subject agreement, the whole predicate followed by dual agreement.

- (44) a. *Dana ale go-co-b go-oo-b esi-a*
man 3.dual hit-ds-3s hit-ds-3s 3.dual-today.past
The two men hit each other
b. *Age eeta eh-i le-ce-b eh-i le-ce-b*
3p yam take-pred go-ds-3s take-pred go-ds-3s
eig-a
3p-today.pst
They took each other yams
c. *Ege tanaw-udo-co-b tanaw-udo-co-b oq-a*
1p pacify-3s-ds-3s pacify-3s-ds-3s 1p-today.past
We made peace with each other

Of more import to us are cases where copying seems to have more of a syntactic or structural function rather than a direct semantic interpretation: We note two types of case. The first, from Mandarin Chinese (Li and Thompson, 1981, Ch 13): in certain sentences of the form Subject+TV+DO+Adverb, where TV is a transitive verb and the DO (direct object) is either indefinite or at least not both referential and animate, the transitive verb must be repeated.

- (45) a. **tā niàn - shū de hěn kuài*
3s read - book prt very fast
S/He reads very quickly
b. *tā niàn - shū niàn de hěn kuài*
3s read - book read prt very fast
S/He reads very quickly

- (46) a. *wǒ pāi - le shǒu liǎng cì
 1s clap perf hand two time
 I clapped (my) hands twice
- b. wǒ pāi - le shǒu p=ai liǎng cì
 1s clap perf hand clap two time
 I clapped (my) hands twice

The second case seems widespread in W. Africa. (47) is from Yoruba (Nigeria; Kobele, 2008) and (48) from Wolof (Senegal; Torrence and Tamba, 2014).

- (47) a. Jimo ɔ ra adiɛ
 Jimo hts buy chicken
 Jimo bought a chicken (hts = high tone syllable)
- b. Rira ti Jimo ɔ ra adiɛ
 red.buy rel Jimo hts buy chicken
 The fact/way Jimo bought a chicken / Jimo's buying a chicken
- c. Rira adiɛ ti Jimo ɔ ra adiɛ
 Red.buy chicken rel Jimo hts buy chicken
 The fact/way Jimo bought a chicken / Jimo's buying a chicken

We note that in the (b) and (c) sentences the verb *ra* is copied to the front and itself reduplicated by the formula $CV \Rightarrow CiCV$, (C = consonant, V vowel). See also Aboh and Dyakonova (2009).

Similar examples are laid out in Kandybowicz (2008) for Nupe.

- (48) a. Ayda teg-na xaaɓis ci taabal
 Ayda put-fin.3s money Prep table
 (Wolof; Senegal)

Ayda put money on the/a table

- b. teg bi Ayda teg xaaɓis ci taabal dafa
 put rel/Comp Ayda put money Prep table focus
 bette
 surprising
 The fact that Ayda put money on a table is surprising

e shǒu liǎng cì
perf hand two time
hands twice

shǒu p=ai liǎng cì
rf hand clap two time
hands twice

spread in W. Africa. (47) is from Yoruba
d (48) from Wolof (Senegal; Torrence and

adiē
chicken
icken (hts = high tone syllable)

o o ra adiē
o hts buy chicken
o bought a chicken / Jimo's buying a

ti Jimo o ra adiē
rel Jimo hts buy chicken
o bought a chicken / Jimo's buying a

(c) sentences the verb *ra* is copied to
ted by the formula $CV \Rightarrow CiCV$, ($C =$
Aboh and Dyakonova (2009).
out in Kandybowicz (2008) for Nupe.

caalis ci taabal
money Prep table

the/a table

da teg xaaalis ci taabal dafa
da put money Prep table focus

put money on a table is surprising

c. teg xaaalis ci taabal bi Ayda teg xaaalis
put money prep table rel/Comp Ayda put money
ci taabal dafa bette
prep table foc surprising

The fact that Ayda put money on the table is surprising

We see then that cross linguistically copying is a widely used process in forming novel utterances, though of limited utility in English. (Occasionally repetition indicates intensity—*My love is like a red red rose*, *He talked on and on*, and a curious usage seems to be growing involving noun repetition: “I don’t mind a little consulting work, but I’d rather have a JOB job”.)

Worth noting too is the other side of the copy coin. Just as CFGs cannot naturally enforce copying, so they cannot not naturally block it either—which is overall good, as a given string can often be used twice in a row with an independent contribution to the meaning in each case. Recall cases like *the mother of the mother of the President*. But in cases with coordination of DPs it seems worse than just redundant (as we have previously noted): *He visited the White House and the Library of Congress* is fine, but **?He visited the Library of Congress and the Library of Congress*, **?He visited neither the Library of Congress nor the Library of Congress* sound a little unearthly. But in a CFG the distribution of all but listably many expressions is decided by their category, not their string part. (*sing*, V) and (*dance*, V) distribute in very similar ways, but are pronounced and semantically interpreted differently.

6.3.3 Discontinuous Constituents

Discontinuous constituents have been illustrated in Chapter 4. And observe:

Fact 3. The parse trees of a CFG satisfy the Exclusivity Condition (Chapter 4: Definition 4.20).

Thus CFGs cannot represent the trees with crossing branches we used for the constituent structure of expressions such as *wrote the names down*, *an easy rug to clean*, I don’t know *which of his students every teacher criticized*. These constitute just a very small sample of expressions with apparent discontinuous constituents. A few others are illustrated in the x' examples below:

- (49) a. [a review of Chomsky’s new book][has just appeared]
a’. a review has just appeared of Chomsky’s new book

- b. [John [has already visited the Louvre]]
- b'. Has John already visited the Louvre?
- c. I [looked up] [the information that was in the phone book]
- c'. I [looked [the information that was in the phone book] up]
- c''. I looked the information up that was in the phone book.

Exercise 6.22. Provide a tree structure for (49c''). It should have two instances of crossing branches.

We might note that linguists are well aware of the discontinuity patterns above. Our point is that to handle them mechanisms must be employed that are not naturally given in a context free format.

Much early work in generative grammar used grammars consisting of two components: a Base and a set of Transformations. The Base was treated often as a CFG. The transformational component took as input the structures determined by the Base and modified them by moving and deleting constituents, thereby providing an account of many apparently discontinuous constituents.

That tradition carries over to current work but with many conceptual changes. Still grammarians would derive many expressions having what we have called discontinuous constituents by starting with ones with no discontinuities and then moving constituents around creating the discontinuity. Our point here is that these mechanisms are not naturally stated in a context free format.

Cross Serial Dependencies

These dependencies have been noted in the grammars of certain Germanic languages (see Shieber, 1985, Manaster-Ramer, 1987 and references cited there). We begin by noting the following structure type in English:

- (50) a. Ann helped Bill walk
- b. Ted persuaded Ann to help Bill walk
- c. Kim helped Ted persuade Ann to help Bill walk

In (50) the grammatical case of subjects is nominative—*she* or *he* in 3rd person singular pronouns, whereas objects of verbs are accusative—*him* or *her*. Pronominalizing the DPs in (50b) yields *He persuaded her to help him walk*. But in German some transitive verbs, like *helfen* 'to help' and *beibringen* 'to teach' take dative case objects (*ihm* in 3rd sg masc.) and others take accusative (*ihn*).

In the English example above the recursion is right peripheral. A rough constituent bracketing of (50c) is:

ady visited the Louvre]]
 dy visited the Louvre?
 e information that was in the phone book]
 ormation that was in the phone book] up]
 ormation up that was in the phone book.
 tree structure for (49c"). It should have two
 hes.

s are well aware of the discontinuity patterns
 handle them mechanisms must be employed
 a in a context free format.
 erative grammar used grammars consisting
 e and a set of Transformations. The Base
 G. The transformational component took
 ermined by the Base and modified them
 stituents, thereby providing an account of
 ous constituents.
 er to current work but with many concep-
 ans would derive many expressions having
 inuous constituents by starting with ones
 hen moving constituents around creating
 t here is that these mechanisms are not
 free format.

s
 n noted in the grammars of certain Ger-
 , 1985, Manaster-Ramer, 1987 and refer-
 by noting the following structure type in

lk
 to help Bill walk
 rsuade Ann to help Bill walk

of subjects is nominative—*she* or *he* in
 whereas objects of verbs are accusative—
 ne DPs in (50b) yields *He persuaded her*
 an some transitive verbs, like *helfen* 'to
 take dative case objects (*ihm* in 3rd sg
 ive (*ihn*).
 ve the recursion is right peripheral. A
 (50c) is:

(51) Kim [helped Ted [persuade Ann [to help Bill [walk]]]]

(51) instantiates basic English word order: SVO (Subject + [Verb + Object]). But consider the pattern in an SOV language. In fact there are several. The one in standard German subordinate clauses, which have SOV order, is seen in (52).

- (52) a. [Der Mann [den Jungen schwimmen] sieht]
 the_{nom} man the_{acc} boy swim_{inf} sees
 The man sees the boy swim
 b. [Sie [den Mann [dem Jungen schwimmen]
 she_{nom} the_{acc} man the_{dat} boy swim_{inf}
 beibringen] sieht]
 teach_{inf} sees
 She sees the man teach the boy to swim

In (52a) *the man* is the subject of *sees* and *the boy* is subject of *swim*, so the pattern in (52) is a self embedding one, as indicated in our bracketing. Nonetheless the grammatical case of *boy* is determined by the final verb. *See* takes accusative, as indicated, but *teach* 'beibringen' takes dative, as illustrated in (52b), which also illustrates a further level of self embedding, 3 case marked DPs followed by three verbs. So this pattern is the $\{a^n b^n \mid n > 0\}$ one. Had the deepest P1, *swim*, consisted of a DP object and a transitive verb, as in *She sees the man teach the boy to paint the house* we would have 4 DPs followed by three verbs, still a self embedding pattern.

Of interest now is that Swiss German (Shieber, 1985) has a verb order which mirrors the DP order, rather than using the self embedding mirror image order. And at least pronoun and proper noun arguments are overtly case marked, helping us identify which DPs are governed by which verbs.

- (53) ...mer em Hans es huus h lfed aastriiche
 ... we_{nom} Hans_{dat} the house_{acc} helped paint
 ... we helped Hans paint the house

The verb *h lfed* 'helped' takes a dative case argument so *em Hans...h lfed* forms a discontinuous constituent. If we replace *h lfed* with *l nd* 'let' then *Hans* must be in the accusative form *de Hans*. If we connect the object DPs in (53) with the verbs that govern their case we see that the dependencies "cross". We indicate this below with indexing:

- (54) ...mer [em Hans]_j [es huus]_k [hålfed]_j [aastriche]_k

Joining co-indexed constituents with lines clearly leads to crossing lines. Finally, Shieber notes that this construction iterates:

- (55) ...mer d'chind em Hans es huus lönd hælfe
 ...we the children_{acc} Hans_{dat} the house_{acc} let help
 aastriche
 paint

... we let the children help Hans paint the house

Dutch exhibits similar cross serial dependencies (Bresnan et al., 1982) but has insufficient case marking on the DPs, so S_s with n (or $n + 1$) DPs followed by n verbs could be derived context freely by self embedding, even though a DP and the transitive verb that governs it would not be generated in the same step, which might make compositional semantic interpretation difficult. For this reason Bresnan et al. suggest that a CFG for Dutch might get the stringset right but not the parse trees needed for semantic interpretation and so fail to be strongly context free.

6.3.4 Necessity: Are All CFLs Possible Natural Languages?

Here our question of concern is whether CFGs are not, in some sense, too strong. Do they allow us to represent properties not present in or not characteristic of NLs? The answer here is a clear yes, but this just means we need constraints on form other than those built into CFGs to characterize NLs. This is hardly surprising, given that CFGs as given have nothing to say about phonological or semantic properties of NL. But as we will see, even at the level of syntax they allow derivational operations that seem decidedly unnatural to the linguist.

As we have seen grammars of languages provide many cases of copying of lexical and phrasal material, albeit with bounds on how complex the copied material can be. But searching grammars we do not find productive morphosyntactic use of palindromes. So speakers do not regularly derive expressions by saying some sequence w and then saying its mirror image. We can, as we have done, find expressions which accidentally have this form, but it is never a regular part of the morphology or syntax of a NL.

A second, more obvious, objection is our earlier remark that any finite set has a CFG. For any such set K let the productions of G be $\{S \rightarrow w \mid w \in K\}$. Then $L(G) = K$. But surely the set of sequences of the word *cat* of length less than 10 billion is not a possible human language. This point is less uninteresting than it seems. What it

es huus]_k [hålfed]_j [aastrîiche]_k

nts with lines clearly leads to crossing lines.
his construction iterates:

em Hans es huus lönd hâlfe
Hans_{dat} the house_{acc} let help

elp Hans paint the house

s serial dependencies (Bresnan et al.,
ase marking on the DPs, so Ss with n
 n verbs could be derived context freely
ugh a DP and the transitive verb that
erated in the same step, which might
interpretation difficult. For this reason
CFG for Dutch might get the stringset
needed for semantic interpretation and
free.

CFLs Possible Natural Languages?

is whether CFGs are not, in some sense,
to represent properties not present in or
e answer here is a clear yes, but this just
form other than those built into CFGs to
lly surprising, given that CFGs as given
onological or semantic properties of NL.
e level of syntax they allow derivational
y unnatural to the linguist.

of languages provide many cases of copy-
rial, albeit with bounds on how complex
But searching grammars we do not find
se of palindromes. So speakers do not
saying some sequence w and then say-
as we have done, find expressions which
t it is never a regular part of the mor-

jection is our earlier remark that any
such set K let the productions of G be
 $= K$. But surely the set of sequences of
an 10 billion is not a possible human
uninteresting than it seems. What it

suggests is that grammars of NLs must have some iterative mechanisms
that enable generating arbitrarily many expressions, even if they get
too long to be usable in practice. So this prompts us to look for the
kinds of recursion characteristic of NLs.

Some more systematic objections concerning the excessive gen-
erative capacity of CFGs are given in the section below on *Closure*
Properties.

Closure Properties

Closure properties are mathematically very helpful ways of character-
izing classes of sets, such as the set of CFLs. As a simple example,
consider how we distinguish even and odd numbers. The even num-
bers are the multiples of 2—numbers of the form $2n$ for some natural
number n . The odd numbers are ones of the form $2n + 1$. Now observe
that the set EVEN of even numbers is closed under addition (+). That
is, for all natural numbers n, m if n and m are both even then so is
 $(n + m)$. By contrast ODD is not closed under addition. For example,
3 and 5 are odd but $3 + 5 = 8$ is not. (Indeed, for all odd n, m we have
that $n + m$ is not odd.) On the other hand, EVEN and ODD are similar
in that both are closed under multiplication.

Now consider the class of CFLs. We have seen that it is closed
under (finite) unions. That is, if U and V are CFLs so is $U \cup V$. $U \cup V$
is often noted $U + V$ in formal language theory. Note though:

Fact 4. The set of CFLs is not closed under arbitrary unions.

Proof. Let K be a stringset which is not context free. For every $w \in K$
define $G[w]$ to be that CFG whose only production is $S \rightarrow w$. So
each $G[w]$ is context free and $L(G[w]) = \{w\}$. But the union of all the
 $L(G[w])$ is K , which is not context free, so the set of CFLs is not closed
under arbitrary unions. $\square$

Earlier we used closure under (finite) union to show that a certain
“tricky” set was in fact context free. And as budding mathematicians
we naturally ask if the set of CFLs is closed under other set theoretical
operations. What about $\cap$ for example? The answer is negative:

Theorem 6.4. *The set of CFLs is not closed under intersection.*

Proof sketch (relies on some earlier unproven claims). We have seen
that $\{a^m b^n c^m \mid n, m > 0\}$ and $\{a^m b^n c^n \mid n, m > 0\}$ are both context
free. But their intersection, $\{a^n b^n c^n \mid n > 0\}$, is not. $\square$

Given a CFG G over a finite V , $\neg L(G)$, the *complement* of $L(G)$
 $=_{\text{def}} V^* - L(G)$. It follows from the facts above that:

Theorem 6.5. *The set of CFLs over a finite V is not closed under complements.*

Proof. $\cap$ is definable in terms of $\cup$ and $\neg$: $X \cap Y = \neg(\neg X \cup \neg Y)$. So if the set of CFLs were closed under $\neg$ it would be closed under $\cap$, but it isn't. $\square$

Theorem 6.6. *The set of CFLs is closed under (finite) cross products. If X and Y are CFLs then so is $X \times Y = \{ww' \mid w \in X, w' \in Y\}$. $X \times Y$ is often noted $X \cdot Y$ in the formal language theory literature.*

Proof sketch. Let G and G' be CFGs generating X and Y respectively. We may assume they have disjoint sets of non-terminals. Let S and S' be the start symbols for G and G' . Form a new grammar G'' with start symbol S'' and whose productions are the union of those of G and G' plus $S'' \rightarrow S S'$. Then $L(G'') = X \cdot Y$. $\square$

Theorem 6.7. *The set of CFLs is closed under star. I.e. if L is context free so is L^* .*

Proof. Let G generate L with start symbol S . Form G' with start symbol S' , not a non-terminal of G . The productions of G' are $S' \rightarrow SS \mid e$ plus those of G . Clearly $L(G') = (L(G))^*$. $\square$

We mention one last closure property which has proven quite useful in linguistic contexts to show that certain stringsets associated with a natural language are not context free. (For further closure properties see Hopcroft and Ullman, 1979, 131–7.) We state the closure condition and then define one new notion used in it.

Theorem 6.8. *The intersection of a CFL with a finite state language is context free.*

The finite state languages are most intuitively defined either as those accepted by finite state machines or those denoted by so called *regular expressions*. (And then called *regular languages*.) Both these notions are defined and studied in the next chapter. Here we give an equivalent definition in terms of the context free apparatus we already have. The definition suffices to make sense of Theorem 6.8, but may seem lacking in intuitive motivation. But it does make explicit certain relations between regular languages and context free ones.

Definition 6.5. A CFG G is *right linear* iff all productions are of the form: $A \rightarrow wB$ or $A \rightarrow u$, where A and B are non-terminals and w, u are terminal strings.

CFLs over a finite V is not closed under

ms of $\cup$ and $\neg$: $X \cap Y = \neg(\neg X \cup \neg Y)$. So
ed under $\neg$ it would be closed under $\cap$, but
□

CFLs is closed under (finite) cross products.
so is $X \times Y = \{ww' \mid w \in X, w' \in Y\}$. $X \times Y$
ormal language theory literature.

be CFGs generating X and Y respectively.
disjoint sets of non-terminals. Let S and S'
and G' . Form a new grammar G'' with start
ctions are the union of those of G and G'
) = $X \cdot Y$.
□

CFLs is closed under star. I.e. if L is

a start symbol S . Form G' with start sym-
of G . The productions of G' are $S' \rightarrow SS \mid$
(G') = $(L(G))^*$.
□

re property which has proven quite useful
that certain stringsets associated with a
text free. (For further closure properties
9, 131–7.) We state the closure condition
on used in it.

ion of a CFL with a finite state language

re most intuitively defined either as those
nes or those denoted by so called *regular*
l *regular languages*.) Both these notions
next chapter. Here we give an equivalent
text free apparatus we already have. The
se of Theorem 6.8, but may seem lacking
it does make explicit certain relations
l context free ones.

right linear iff all productions are of the
ere A and B are non-terminals and w, u

Definition 6.6. The *regular* stringsets are just the $L(G)$ for G a right linear grammar.

Example Write $a(ba)^*$ for the set of strings consisting of an a followed by any number (including zero) of copies of ba . a and $abababa$ are in this set, b and $abab$ are not. Here is a right linear grammar for $a(ba)^*$:

$$S \rightarrow aT, T \rightarrow baT \mid e.$$

Exercise 6.23. Exhibit a derivation of $ababa$ using the grammar above.

Exercise 6.24.

- Complete the following definition: A CFG G is *left linear* iff ____
- Exhibit a left linear grammar for $a(ba)^*$.

Let us see that the image of a CFL under a string homomorphism is itself a CFL. We define:

Definition 6.7. Given vocabularies V and U , a function h from $V^* \rightarrow U^*$ is a *string homomorphism* iff it satisfies conditions (a) and (b) below:

- $h(e) = e$ and
- $h(w) = h(y) \frown h(b)$ if $w = y \frown b$, any $y \in V^*$, any $b \in V$.

Obviously the values of a string homomorphism h as above are given by its values on the elements of V . So to define such a function it suffices to say what its values are at the elements of V . Then we extend h to a function h' from sets of strings over V^* as follows:

Definition 6.8. If h is a string homomorphism from V^* to U^* and K is any subset of V^* , then $h'(K) = \{h(k) \mid k \in K\}$. When no confusion results we write $h(K)$ instead of $h'(K)$.

Theorem 6.9. For G a CFG with terminal vocabulary V , for any string homomorphism h from V^* to any U^* , $h(L(G))$ is context free.

Exercise 6.25. Say informally why the theorem immediately above holds.

Theorems 6.8 and 6.9 are frequently invoked in showing that certain languages we are interested in are not context free. This is so for example for the Bambara claim (Culy, 1985) and the Swiss German claim (Shieber, 1985).

These theorems also enable us to see the interest of working with artificial languages with very small vocabularies. If we start with a language with a large vocabulary (e.g. a model of English), a homomorphism may map the language to one with a very small alphabet by being massively not one to one, e.g. by mapping many of its lexical strings to the same element of U^* (perhaps the ϵ string). Then this very artificial language may be easily proven to be not context free, hence the original one must fail to be context free by the theorems. On the other hand, the closure facts do suggest properties that do not seem characteristic of NLs. We do not expect to be able to paddle up the Amazon and find some tribe whose utterances are exactly those in the cross product of English, Malagasy, Mandarin, Turkish and Yoruba. But as earlier, the problems here may relate in part of ones concerning phonology and semantics.

6.3.5 Inferring Structure from Strings: Normal Form Theorems

We have insisted that evaluating a proposal for the adequacy of a class of grammars as models of NLs involves crucially assessing its strong generative capacity, not just its weak generative capacity, that is, the set of strings it generates. Some results in CFGs serve to highlight this point. It has been useful in developing proofs that certain string sets are, or are not, context free to show that an arbitrary CFG can be transformed to one all of whose rules are in a certain form. Here are two well known useful cases:

(56) Chomsky Normal Form

Given a CFG G we can construct a CFG G' with the property that:

1. $L(G) = L(G')$ and
2. All productions of G' are of one of the two forms:
 - A. $A \rightarrow BD$, where B and D are non-terminal symbols, and
 - B. $A \rightarrow w$, w a terminal string.

(57) Greibach Normal Form

Given a CFG G we can construct a CFG G' such that:

1. $L(G) = L(G')$ and
2. All productions are of the form: $A \rightarrow a\alpha$, where a is a terminal symbol and $\alpha \in (N \cup T)^*$.

able us to see the interest of working with small vocabularies. If we start with a vocabulary (e.g. a model of English), a homomorphism to one with a very small alphabet by one, e.g. by mapping many of its lexical items to U^* (perhaps the e string). Then this can be easily proven to be not context free, and will not be context free by the theorems. On the other hand, it does suggest properties that do not seem to expect to be able to paddle up the shore whose utterances are exactly those in the languages, Mandarin, Turkish and Yoruba. There may be relations in part of ones concerning

from Strings: Normal Form

ing a proposal for the adequacy of a class of languages involves crucially assessing its strong and weak generative capacity, that is, the results in CFGs serve to highlight this. Developing proofs that certain string sets cannot be shown that an arbitrary CFG can be reduced to one whose rules are in a certain form. Here are

construct a CFG G' with the property

are of one of the two forms:

B and D are non-terminal symbols,

terminal string.

construct a CFG G' such that:

the form: $A \rightarrow a\alpha$, where a is a terminal symbol and $\alpha \in (N \cup T)^*$.

Now considered jointly these two theorems tell us that merely given the set of strings a grammar generates is not sufficient to determine the form of the rules of the grammar (even limiting them to context free ones) as the form of the rules in Chomsky Normal Form and in Greibach Normal Form are not the same.

We turn now to the study of a more restrictive class of grammars, those of the regular or finite state languages. The stringsets they can parse are a proper subset of those parsable by context free grammars. Nonetheless grammars of this form have proven useful as morphological parsers, even for languages with elaborate and complex morphology (Hankamer, 1986 and Creider et al., 1995, and references cited there.)

Addendum 1. How to convert $L(G)$, G context free, into an inductive (bottom up) definition by recursion.

Let $G = \langle N, T, S, \rightarrow \rangle$ be context free. We show how to define $L(G)$, the language generated by G , in an ordinary bottom up way, one to which our ordinary proofs by recursion apply.

To facilitate the conversion we define an *unconstrained grammar* G to be a four-tuple: $\langle \text{Cat}_G, V_G, \text{Lex}_G, \text{Rule}_G \rangle$, as follows (omitting subscripts when no confusion results.) Elements of Cat_G are called *grammatical categories*, elements of V_G are called *phonological strings*. A *possible expression* is an ordered pair (s, C) , where s is a phonological string and $C \in \text{Cat}$. We note this set PE_G , the set of *possible expressions* of G . Formally PE_G just abbreviates $(V_G^* \times \text{Cat}_G)$. Lex_G is finite subset of PE_G called the *Lexicon* of G , and Rule_G is a finite set of partial functions from PE_G^* to PE_G . So Lex_G is just a finite set of possible expressions, and an $F \in \text{Rule}_G$ is simply a partial function from sequences of possible expressions to possible expressions. Then of course we define $L(G)$, the *language* generated by G , to be the closure of Lex_G wrt Rule_G .

Given a CFG $G = \langle N, T, S, \rightarrow \rangle$ we build an unrestricted G as follows: $G = \langle N, T, \text{Lex}_G, \text{Rule}_G \rangle$ where:

- Step 1: $\text{Lex}_G = \{ \langle w, A \rangle \mid A \rightarrow w \text{ in } G, \text{ where } w \in T^* \}$ and
- Step 2: Each production P of G not of the form $A \rightarrow w$ as above has the following form:

P is: $A \rightarrow w_0 A_1 w_1 \dots w_n A_n w_{n+1}$, where the $A_i \in N$ and the $w_j \in T^*$

NB: The w_i here are fixed terminal strings, like *neither* and *nor* in a rule like: $\text{VP} \rightarrow \text{neither VP nor VP}$, used in deriving *neither laughed nor cried*.

For each such P we define $F(P)$ in Rule_G as below, and then set Rule_G to be $\{ F(P) \mid P \text{ a non-lexical production of } G, \text{ as above} \}$.

- a. The domain of $F(P)$ is $\{(u_1, A_1), \dots, (u_n, A_n) \mid u_i \in T^*\}$ and
- b. $F(P)$ maps each n -tuple $(u_1, A_1), \dots, (u_n, A_n)$ in its domain to $w_0 u_1 w_1 \dots w_n u_n w_{n+1}$.

By way of example, consider the CFG G given by: $S \rightarrow ab, S \rightarrow aSb$. Converting it to an unrestricted grammar G' gives us a Lexicon $\{(ab, S)\}$ and the single rule F with domain $\{(u, S) \mid u \in \{a, b\}^*\}$ given by: $F((u, S)) = (a \frown u \frown b, S)$.

To prove that $L = \{(a^n b^n, S) \mid n > 0\} \supseteq L(G)$ we note that $(ab, S) \in L$ and L is closed wrt F : if $(w, S) \in L$ then for some k , $w = (a^k b^k, S)$, so $F(w, S) = (aa^k b^k b, S) = (a^{k+1} b^{k+1}, S) \in L$. Thus L is one of the sets we took the intersection of in forming $L(G)$. Going the other way show that $K = \{n \in \mathbb{N} \mid n > 0 \Rightarrow (a^n b^n, S) \in L(G')\} = \mathbb{N}$, so for all n , $(a^n b^n, S) \in L(G')$. Clearly $0 \in K$ (vacuously) and $1 \in K$, and $n+1$ is if n is, so $\mathbb{N} \subseteq K$, whence equality holds. Thus we see that w is derivable from S in G iff (w, S) is an element of $L(G')$. $\square$