Appendix A

Composition and Decomposition of Patterns

A.0 Introductory Notes

The aim of this appendix is to sketch pattern matching analysis in suitable detail to supplementing discussions presented in Chapters 2, 3. Certain assumptions will be revised and discussed in greater detail.

This appendix comprises nine sections. Section A.1 gives a reappraisal of some assumptions made in PMA, trying to contrast pattern matching analysis with more popular phrase structure analysis. Sections A.2 and A.3 supplement treatments in Chapters 2 and 3, and offer details of pattern composition and pattern decomposition. Section A.4 provides more technical details of pattern composition and decomposition. Section A.5 discusses, from a PMA perspective, conditions for syntax to "emerge". Section A.6 discusses how PMA recognizes effects of **phrases**. Section A.7 discusses how PMA deals with morphological phenomena. Section A.8 discusses how the notion of subpatterns differs from the notion of **subcategorization frame**. Section A.9 concludes this appendix.

A.1 Reappraisal of Pattern Matching Analysis

This section discusses crucial assumptions of pattern matching analysis in great detail.

A.1.1 Characteristics of pattern matching analysis

Pattern matching analysis is not only a **compositional** theory of language syntax (with implicit internal combinatorial semantics); it is also a **constructivist** theory, if not a "constructionist" one.[1] By a *constructivist* model, I mean a model in which all constructs are inductively constructed from a finite set of "primitives". Indeed, pattern matching analysis tries to define all constructs inductively. This implies that the theory always tends to generate less than necessary, and will be under the threat of **undergeneration** more than **overgeneration**, thereby forbidding use of "output filters" and the like.

Pattern matching analysis is by no means a **derivational** theory, at least in a significant sense. One must be careful, though. It is difficult to claim that a theory is not a derivational theory in strict sense, because the role(s) of derivation in a derivational theory must have counterparts in any theory that its proponents claim to be nonderivational. After having noting this, I suggest that most "effects" attributed to derivations are reinterpreted in PMA as effects of pattern composition, and for this reason derivation is made implicit and will play no major role.

Adopting a connectionist philosophy, the proposed framework recasts, in a sense, the nature of the question of what is knowledge of language. It states, along with Hudson's word grammar (1984, 1990), and Langacker's cognitive grammar (1987, 1991*a*,*b*), that **knowledge of language**, if any, **is nothing but knowledge of words**, granted that words (and constructions) are not anything but what are called subpatterns.

Patterns are **composed out of smaller patterns**, or **subpatterns**. I will call this such operation (and process) **pattern composition (out of subpatterns)**. Pattern composition may correspond to **generation** in the technical sense of generative linguistics. But I will not discuss here whether pattern composition, in the proposed terminology, and generation are equivalent notions, since it is immaterial.

A.1.2 Remarks on the notion surface form(ation)

Like many approaches stemming from generative grammar, pattern matching analysis investigates properties of "abstract" objects, which I will **syntactic patterns**. It distinguishes syntactic patterns from **surface forms** or better put **surface formations**. PMA posits, for example, (1)b as the syntactic pattern that accounts for surface form(ation) (1)a:

(1) a. *What do you think is waiting for you?*
b. *what do you think Ø is waiting for you*

I note that surface form(ation)s like (1)a are not a real object of linguistic inquiry. Note that it is not clear at all **what** (1)a **represents**. If is true that (1)a represents what one calls a "sentence", but what is a sentence?

To answer this, PMA assumes that (1)a represents, in terms of orthography, **what** (1)b **represents**. In this assumption, PMA claims that if either (1)a or b is fallacious, it is not (1)b but (1)a. This assumption may be surprising, but is consistent with all facts that we will treat. Thus, it is a fiction that form(ation)s like (1)a exist.

Formations like (1)b, called syntactic patterns, are abstract objects in that they are real only mentally. They are as much abstract as circles, triangles, faces of your friends, and all mathematical structures. Their abstractness can be shown by representing them in more adequate form. (1)b, for example, is more adequately represented in either of the following:

(2) a. *what ... do ... you ... think ... Ø ... is ... waiting ... for ... you*
b. *what < do < you < think < Ø < is < waiting < for < you*

Here, symbols "..." and "<" denote the precedence operator.

To make essential properties of precedence relation clearer, I will discuss below its relevant properties, thereby introducing the notion of **minimum syntax (for strings)**. This is done to show that it is reasonable to view syntax as a **system of co-occurrences**, which is defined by making reference to **precedence** alone.

A.1.3 The notion of "minimum syntax" for strings

To begin with, we define the binary relation precedence, denoted by $a < b$ (and ... a ... b ...) as follows:

(3) **Definition.** We write $a < b$ iff a precedes b. Precedence relation is irreflexive, transitive, and not symmetric.

Incidentally, it is useful to extend < to the relation of **overlap**, denoted by $a \not\preccurlyeq b$. Note that a B iff $B = ac$ ($a = B$ if $c = \lambda$). Note also that ab bc if $a < c$ ($ab = bc$ if $a = c = \lambda$).

Next, we define a **precedence condition** based on a set of precedence relations, as follows:

(4) **Definition.** A precedence condition, $a_1 < a_1 < ... < a_n$, is a set of precedence relations, all of which are satisfied.

Remarks. Precedence condition $a < b$ describes all strings of the form $XaYbZ$, where $X, Y, Z \in V^*$ ($a, b \in V$), with empty symbol λ included in V.

Next, we define weak interpretation of a string as follows:

(5) **Definition.** A weak interpretation of a string is a precedence condition.

To supplement this definition, we defined strong interpretation of a precedence condition as follows:

(6) **Definition.** A strong interpretation of a precedence condition is a string.

Example 1. The weak interpretation of string *abcd* is precedence condition $a < b < c < d$ (= a ... b ... c ... d).

Example 2. The strong interpretation of precedence condition $a < b < c < d$ (= a … b … c … d) is string *abcd*.

Remarks. A strong interpretation of a precedence condition must be margin-free; in other words, all occurrences of "…" have to be eliminated.

We define exhaustive (discontinuity-proof) segmentation of a string, based on the weak interpretation of it.

(7) **Definition.** Exhaustive segmentation of a string S is the partially ordered set (V, <), whose top element is the weak interpretation of S.

Example. $a < b < c < d$ is the top element of the partial order set $G = (S, <)$, where S designates the set of segmentations of *abcd.*

(8)
4. $a < b < c < d$
3. $a < b < c, a < b < d, a < c < d, b < c < d$
2. $a < b, a < c, a < d, b < c, b < d, c < d$
1. a, b, c, d
0. ø

The Hasse diagram in Figure A.1 illustrates the lattice structure of G.

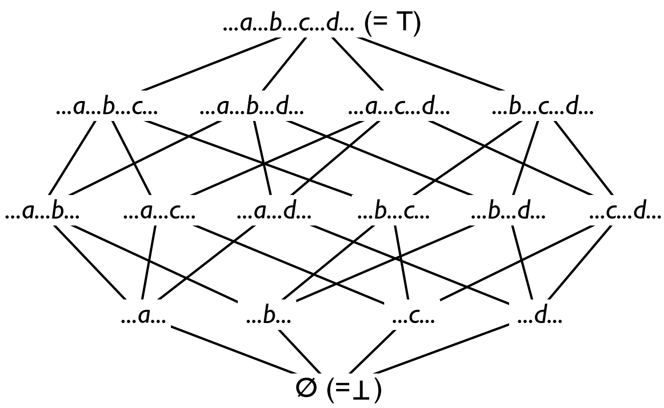


Figure A.1

The diagram here illustrates the exhaustive segmentation of *abcd.*

Finally, we define the minimum syntax for strings as follows:

(9) **Definition.** The minimum syntax for string S is the weakest substructure in the exhaustive segmentation of S, with some measure to determine which substructure is the weakest.

Remarks. There may be cases where the weakest substructure is the same as the exhaustive segmentation. For example, the minimum syntax for *ab* is the lattice structure: {...*a*...*b*... (top), ...*a*..., ...*b*..., ø (bottom)}.

A.2 Pattern Matching Analysis in Contrast with Phrase Structure Analysis

In this section, I will compare pattern matching analysis with standard phrase structure analysis.

A.2.1 Review of phrase structure analysis

To begin with, consider the following:

(10) *Bill undergoes an operation.*

Pattern matching analysis assumes that surface linguistic forms like this are surface formations that can be equated with patterns, composed out of subpatterns. But what are patterns and subpatterns? Unfortunately, we are not ready to answer this question immediately. So, let us briefly review how popular accounts go.

It is commonplace to think that form in (10) is made from four words, as follows:

(11) w_1 = *Bill*,
w_2 = *undergoes* (< *undergo*),
w_3 = *an*,
w_4 = *operation*

Furthermore, most syntactic theories posit that the form in (10) is a **concatenation** of the four words, as in the following form:

(12) $S = w_1 + w_2 + w_3 + w_4$

Here, S stands for "sentence". Symbol "+" designates the concatenation operator. Concatenation is asymmetric ($a + b \neq b + a$).

In generative linguistics, objects like that in (12), which are called **strings**, are structures that are (indirectly) generated by a special device called the **base component**, which is very much like a **production system** in the sense of E. Post (1943). Note that (12) can be identified with $w_1w_2w_3w_4$ which is given as a string of **terminals** of a phrase marker generated by production rules in (13).

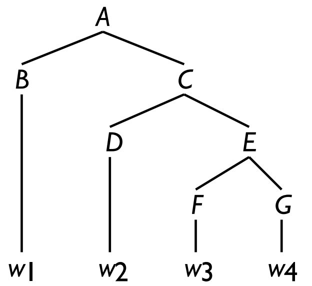
(13) i. $A \to BC$, $C \to DE$, $E \to FG$;
ii. $B \to w_1$, $D \to w_2$, $F \to w_3$, $G \to w_4$

The leftmost derivation of $w_1w_2w_3w_4$ from the initial symbol A is:

(14)
$$
\begin{aligned}
& A \\
\Rightarrow\ & BC \\
\Rightarrow\ & w_1\ C \\
\Rightarrow\ & w_1\ DE \\
\Rightarrow\ & w_1\ w_2\ E \\
\Rightarrow\ & w_1\ w_2\ FG \\
\Rightarrow\ & w_1\ w_2\ w_3\ G \\
\Rightarrow\ & w_1\ w_2\ w_3\ w_4
\end{aligned}
$$

This operation gives a "derivation tree" represented by (15)b, or (15)a for expository purposes.

(15) a. $[_A\ [_B\ w_1\ ][_C\ [_D\ \ w_2\ ][_E\ [_F\ w_3\ ][_G\ w_4\ ]]]$
b.



Preterminal symbols, A, B, ..., G, are arbitrary. "Labels" assigned by a linguistic theory to A, B, ..., G do not affect the validity of the argument.

A.2.2 Pattern composition

Instead of appealing to the kind of structures in (15)a, b, PMA posits that objects like (10) are equated with **(syntactic) patterns** obtained by **superposition** of **subpatterns** u_1, u_2, u_3, and u_4, whose details are schematically represented in (16), where u_i denotes a subpattern ($u_i \neq w_i$), symbol "×" denotes the superposition operator ($a \times b = b \times a$).

(16)
$$
\begin{aligned}
P &= u_1 \times u_2 \times u_3 \times u_4 \\
&= \mathscr{C}(U), \text{ where } U = \{u_1, u_2, u_3, u_4\}
\end{aligned}
$$

Here, u_i is a subpattern. P is a pattern composed of subpatterns.

The appeal of representation in (16) is that superposition, whose operator is denoted by ×, is strictly **ordering-free**. This point should be clear from the fact that the operand of $\mathscr{C}$ is a set (e.g., $U = \{u_1, u_2, u_3, u_4\}$) rather than an n-tuple (e.g., $<u_1, u_2, u_3, u_4>$).

It should be mentioned that the utility of ordering-freeness trades off with another. It is assumed that every subpattern is like a string. This is possible only under a radical reinterpretation of so-called lexical items specified below.

A.2.3 Structure of subpattern

Proposed reinterpretation of lexical items is this: instead of conceiving words as unstructured, decontextuated items like $w_1, \ldots, w_4$ defined in (11), PMA conceives of them as structured units $u_1, \ldots, u_4$ in the following way:[2]

(17) u_1 = ***Bill*** *V (O)*[3]
u_2 = *S* ***undergoes*** *O*
u_3 = *S V* ***an*** *(AdN) N*[4]
u_4 = *S V (D)* ***operation***[5]

For expository purposes, I shall call units like $u_1, \ldots , u_4$ in (17) **contextuated units**, contrasted with $w_1, \ldots , w_4$ in (11), which may be called **decontextuated units**.

A.2.4 *Anchors* and *Glues* — components of a subpattern

In (17), and elsewhere, I will appeal to a notational convention which plays a crucial role in the proposed framework, defined as follows:

(18) i. In bold (italic) face are "substantial" subcomponents of patterns, to be called **anchors** of (sub)patterns. For example, ***Bill*** is the anchor of *Bill* = ***Bill*** *V (O)* (≠ *S V* ***Bill*** which encodes *Bill* as an object).
ii. In normal (italic) face are "relational" subcomponents, to be called **glues** of (sub)patterns. For example, *V* and *(O)* in *Bill* = ***Bill*** *V (O)*, and *S* and *O* in *undergo* = *S* ***undergo*** *O* are all glues.

Units like those in (17) are contextuated exactly because any of such units has at least one glue. Any unit consisting of an anchor (or anchors) surrounded by glues are called **syntactic patterns**, or simply **patterns**.

Glues play a special role in PMA description of natural language syntax. As I will discuss later, glues need not be such variables as *S*, *V*, *O*. They can be itemic units called "shadows", like the shadow *came back* in $Bill_1$ = ***Bill*** *came back*, $Bill_2$ = ***Bill*** *drank too much*, and so on. Crucially, schematic glue *V* in this case is a **generalization of contexts** ... *came back*, ... *drank too much*, etc. Details will be discussed in Appendix B.

A.2.5 Connectionist root of the notion of contextuated units

The idea of contextuated units such as $u_1, \ldots, u_4$ in (17) is inspired by the idea of **wickelphones**, which Rumelhart and McClelland (1986) used to "teach" their connectionist networks phonology of English. As I will discuss in Appendix B, wickelphones are context-sensitive units of the form *yxz*. Technically, this can be seen as an "allophone" of *x*, that occurs in, and therefore accommodated to, the context of *y* __ *z*, where *y* and *z* are variables for phonological unit. Thus, /kæt/[6] (for *cat*) is an allophone of /æ/ such that it is contextuated by /k/ at left and /t/ at right.[7] Wickelphonology of /#**k**æt#/ is thereby given as a set {#**k**æ, kæt, æt#}. For more details, see Appendix B.

A.2.6 Composing subpatterns into a single pattern by superposition

How are patterns like those given in (17), if well defined, composed without making use of derivation trees? The PMA position is clear and simple: composition of patterns, understood here to be superposition, can be carried out by making use of their "overlaps".

Leaving technical details for discussions in Section A.5, let it suffice here to introduce a scheme for representation such as in (19), where four subpatterns 1, 2, 3, and 3 are **synchronized** along time to produce pattern 0, assuming that columnwise (vertical) unification is **superposition.**

(19)	0.	***Bill***	***undergoes***	***an***	***operation***
		⇑	⇑	⇑	⇑
	1.	***Bill***	V_1	$(O_{1.1}$	$O_{1.2})$
	2.	S_2	***undergoes***	$O_{2.1}$	$O_{2.2}$
	3.	S_3	V_3	***an***	N_3
	4.	S_4	V_4	(D_4)	***operation***

Here and elsewhere, *S* encodes subject, *O* encodes object, *N* encodes (head) noun, and *D* encodes determiner. Note that specifications in 1, 2, 3 and 4 in (19) are identical with u_1, u_2, u_3 and u_4 in (17).

I will refer to (19) as a **composition/decomposition table** (C/D table short). It is designed to facilitate seeing how subpatterns are composed into a pattern, on the one hand, and how a pattern is decomposed into subpatterns, on the other.

A C/D table comprises two components. One is a component called the **base pattern**, which is at 0. Another is a component called a **co-occurrence matrix**, which constitutes an $n \times n$ matrix. In the C/D table in (19), thus, the base is pattern 0 = ***Bill undergoes an operation***. The co-occurrence matrix is the matrix made of four subpatterns 1 = ***Bill*** *V (O)*, 2 = *S* ***undergoes*** *O*, 3 = ***an*** *N* and 4 = *(S) V (D)* ***operation***.

The relation of the co-occurrence matrix to the base in (19) is indicated by arrows ⇑: base pattern 0 is composed of its subpatterns 1, 2, 3 and 4. Columnwise unification is the way to compose subpatterns. Thus, a more exact form of pattern 0 can be given:

(20) 0. $\textbf{\textit{Bill}}\times S_2\times(S_3)\times S_4 < \textbf{\textit{undergoes}}\times V_1\times(V_3)\times V_4 < \textbf{\textit{an}}\times(N_3) < \textbf{\textit{operation}}\times O_1\times O_2\times N_3$

Here, "<" denotes the operator of precedence relation. The operator of unification is denoted by "×". I assume that composition is stronger than precedence.

It is possible (and perhaps quite reasonable) to reinterpret the operator of concatenation in terms of precedence by identifying "+" as "<". More details of pattern composition will be discussed in the next section.

A.3 Subpatterns Emerging through Schematization

Syntagmatic informations of word schemas, encoded in horizontal mode, are exactly what one needs in doing syntax. It may be asked, thus, Where do such informations come from? This question will be addressed in this section.

A.3.1 Decomposition of a pattern by diagonalization

How to do to obtain subpatterns in (17), reproduced here with slight modification, to account for the syntax of *Bill undergoes an operation* [= (10)]?

(17)
1. ***Bill*** *V (O)*
2. *S* ***undergoes*** *O*
3. *S V* ***an*** *(AdN) N*
4. *S V (D)* ***operation***

Adequately idealized, an essential part of syntactic analysis necessitates a procedure by which, the co-occurrence matrix in (21) is converted into the one in (22).

(21)

1.	***Bill***	*undergoes*	*an*	*operation*
2.	*Bill*	***undergoes***	*an*	*operation*
3.	*Bill*	*undergoes*	***an***	*operation*
4.	*Bill*	*undergoes*	*an*	***operation***

(22)

1.	***Bill***	*V*		*(O)*
2.	*S*	***undergoes***		*O*
3.	*S*	*V*	***an***	*N*
4.	*S*	*V*	*(D)*	***operation***

Simply put, specifications in (21) are **generalized** and **schematized** to reduce redundancy in them.

The procedure required implies that syntactic analysis is partly a special kind of categorization task through which context-sensitive categorizations such as follows are achieved:

(23) i. *Bill* ← *S/ __ undergoes an operation*
ii. *undergoes* ← *V/ Bill __ an operation*
iii. *an operation* ← *O/ Bill undergoes __*
iv. *an* ← *D/ Bill undergoes ___ operation*
iv′. *an* ← *D/ ___ operation*
v. *operation* ← *N/ Bill undergoes an ___*
v′. *operation* ← *N/ an ___*

Here, the operator of categorization is indicated by "←". Note that each categorization consists of an lexical item and its (local) context. To state this in a general manner, it is helpful to appeal to the following notation.

(24) $x \leftarrow K/\xi(u)$

This means that lexical unit x is categorized as K in context $\xi(x)$. This indicates that x's identity in terms of its category is given as an ordered pair $<x, \xi(x)>$. Thus, it is clear that syntactic theory must formalize contexts.

The interdependence of units and their contexts in categorization procedure can be better characterized by introducing the notion of **pattern diagonalization**. Suppose an abstract matrix, as follows, is given where all $r_{i,j}$ (except $i = j$) is not lexically determined.

(25)

0.	***Bill***	***undergoes***	***an***	***operation***
1.	***Bill***	$r_{1,2}$	$r_{1,3}$	$r_{1,4}$
2.	$r_{2,1}$	***undergoes***	$r_{2,3}$	$r_{2,4}$
3.	$r_{3,1}$	$r_{3,2}$	***an***	$r_{3,4}$
4.	$r_{4,1}$	$r_{4,2}$	$r_{4,3}$	***operation***

Note that the base determines the maximal range of contexts for subpatterns.

Given (25) as the initial state, the syntactic analysis for (22) consists in the discovery of **dependencies** based on pairwise interactions. Categorization is successful if the following dependencies are discovered:

(26) i. $r_{1,2} \leftarrow V$; $r_{1,3} \leftarrow (O_1$; and $r_{1,4} \leftarrow O_2)$:
in words, relative to the 1st unit, *Bill*, the 2nd unit is *V*; the 3rd and 4th units are the first and second "segments" of an optional O of the 1st unit.

ii. $r_{2,1} \leftarrow S$; $r_{2,3} \leftarrow O_1$; and $r_{2,4} \leftarrow O_2$:
in words, relative to the 2nd unit, *undergoes*, the 1st unit is *S*, and the 3rd and 4th units are the first and second segments of its obligatory *O*.

iii. $r_{3,1} \leftarrow S$; $r_{3,2} \leftarrow V$; and $r_{3,4} \leftarrow N$:
in words, relative to the 3rd unit, *an*, the 1st and 2nd units are *S* and *V*; and the 4th unit is *N*.

iv. $r_{4,1} \leftarrow S$; $r_{4,2} \leftarrow V$; and $r_{4,3} \leftarrow D$:
in words, relative to the 4th unit, *operation*, the 1st and the 2nd units are *S* and *V*, and the 3rd unit is *D*

Under this interpretation, the task in question is a function that "reduces" binary relations in abstract matrices like (25) by checking all the binary relations one by one. It seem clear that this kind of task could not be efficiently calculable unless it is carried out by parallel computational devices such as neural networks.

A.3.2 From *itemic* to *schematic* encoding

It is clear that the kind of encoding as in (21), which, for convenience, I will call **token-based encoding**, or interchangeably **itemic encoding**, is not fully useful. Itemic encoding is too "specific" to capture interesting generalizations. Indeed, this deficiency was severely criticized by those antagonists of connectionism like Fodor and Pylyshyn (1988), and Pinker and Prince (1988).[8]

Itemic encoding is merely the most obvious way of making use of context-sensitivity, and by no means is it the only one. There is indeed another, quite useful encoding scheme. The only required revision is simply to allow $r_{i,j}$ to be a variable such as *S*, *O*, *V*. This is what (22) introduces, to which I will refer as **type-based encoding**, or interchangeably **schematic encoding**.

The procedure that converts, by schematization, a token-based encoding into a type-based encoding is essential from the perspective of language learning. If such procedure is guaranteed, then it disproves the "unlearnability" thesis of grammar.

Encouraged by connectionist results of Elman (1990, *et seq.*), PMA claims that such procedure is guaranteed by the mechanism of the brain, and thereby claims that subpatterns "emerge" through statistically based generalizations over co-occurrences of words. In this sense, there should be a "discovery procedure", in the sense of structuralist linguistics, endorsed by a straightforward learning algorithm that is connectionistically realizable.

Co-occurrence matrix may serve as a key to describe the lexical-grammatical structures that the neurally implemented discovery procedure brings. To make this guess more plausible, let me give more detailed arguments.

A.4 More on Pattern Composition and Decomposition

In this section, we will be concerned with more technical details of pattern composition and decomposition.

A.4.1 Details of co-occurrence matrix

Co-occurrence matrix is so-called because it always takes the form of an $n \times n$ matrix, given base patterns comprise n units.

In co-occurrence matrices, the i^{th} row encodes the i^{th} unit of the "base" form. More specifically, pattern 0 in (19) is a superposition of subpatterns 1, 2, 3, and 4 if and only if:

(27)
i. Anchor ***Bill*** matches *S*-glue of u_2 = *S* ***undergoes*** *O*,
ii. Anchor ***undergoes*** matches *V*-glue of u_1 = ***Bill*** *V*,
iii. Anchor ***an*** matches *D*-glue of u_4 = *V (D)* ***operation***,
iv. Anchor ***operation*** matches *N*-glue of u_3 = ***an*** *N*, and *O*-glue of u_2 = *S* ***undergoes*** *O*,

Pattern matching analysis is so-called because the notion of pattern matching plays a crucial role as specified.

A.4.2 Pattern matching as "relaxation"

I interpret that pattern matching is a **relaxation** in the sense of Arbib (1989), under the interpretation that subpatterns express multiple constraints to be "relaxed".

A.4.3 Syntax encoded in vertical and horizontal modes

Pattern composition crucially relies on overlaps among subpatterns. Pattern composition is in vertical mode, in that pattern matching takes place vertically. In addition, there is a horizontal mode. In horizontal mode, each word schema is understood as a "declarative" statement of co-occurrence restrictions (hence the name co-occurrence matrix). To be more explicit, patterns in (19) specify, in horizontal mode, that:

(28)
i. Subpattern 1 states that *Bill* precedes a main verb, *V*.
ii. Subpattern 2 states that *undergoes* postcedes *S* and precedes *O*.
iii. Subpattern 3 states that *an* postcedes *S V* and precedes *N*.
iv. Subpattern 4 states that *operation* postcedes *V*.

Note that in those co-occurrence statements, adjacency is not encoded. So, the link from ***Bill*** to *V* is infinitely stretchable, at least theoretically.

A.4.4 Words as schemas encoding precedence-sensitive dependency

Patterns in the co-occurrence matrix are words; but they are not mere words. As we have seen in Chapter 2, they are better characterized as **schemas** not (only) in the sense usually assumed in cognitive linguistics literature but (also) in the sense that neural cognitive scientists like Arbib (1989), and Arbib, Hill, and Conklin (1987) assume. So, it is not pointless to state that given subpatterns are roughly words, subpatterns are **word as schemas**, or simply **word schemas**.[9]

What do these schemas encode? My interpretation is that they encode **precedence-sensitive dependency** (or dominance) in that they are themselves declarative statements of co-occurrence restrictions.

Informations encoded horizontally are exclusively syntagmatic, correlated with relational concepts. But, as far as I can see, there is no evidence that they have a conceptual basis. In asserting this, I am deviating from the trends of cognitive linguistics which attempt to "spatialize" syntax. For details of the move, see Lakoff's (1987: 283) "spatialization of form hypothesis" and Deane's (1992) development of the hypothesis, and Langacker's (1991*a*) "grammar as image".

A.4.5 Diagramming co-occurrence matrix

Technically, co-occurrence matrices like (19) are optimizations of more abstract $n \times n$ matrices, which are mechanically obtained by **diagonalization**.

(29)
$$\begin{array}{llllll}
0. & 1 & 2 & \cdots & n \\
1. & r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\
2. & r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
n. & r_{n,1} & r_{n,2} & \cdots & r_{n,n}
\end{array}$$

This matrix encodes an array of pairwise relations $r_{i,j}$ of the i^{th} unit to the j^{th} with relational label r.

Under this definition, it is easy to see that co-occurrence matrix can be diagrammed in a straightforward manner by means of **precedence/dependence diagram**, or **communication diagram**:

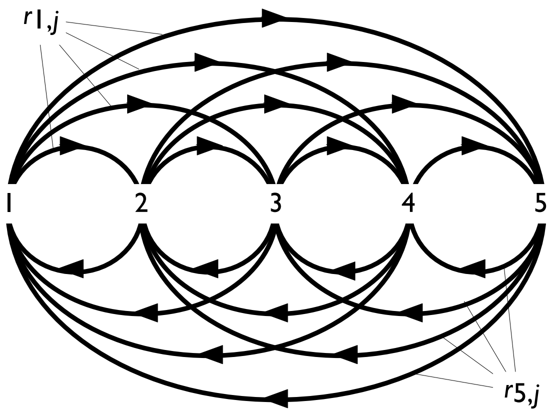


Figure A.2

Illustrated here is a communication diagram for the 5×5 co-occurrence matrix for **12345** (e.g., [$_1$ ***Bill***][$_2$ ***threw***][$_3$ ***his boots***][$_4$ ***at***][$_5$ ***the door***]).

My interpretation of diagrams like the one above is that there are two different kinds of relations encoded. LR arrow (⇒) encodes a **demanding relation** in which the i^{th} unit demands the j^{th} to be there with relational label R. Similarly, RL arrow (⇐) encodes a **supporting relation** in which the j^{th} unit supports the i^{th} to be there if the i^{th} bears relational label R.

Some other relevant properties illustrated by this diagram are:

(30)
i. Links in the upper half of this diagram correspond to $r_{i,j}$ ($i < j$) in the upper right triangle of M
ii. Links in the lower half of this diagram correspond to $r_{i,j}$ ($i > j$) in the lower left triangle of M

For clarity, examine properties of the most basic pattern, *S V O*, by way of the equivalence.

Diagonalization of ***S V O*** gives a 3×3 co-occurrence matrix as follows, where $\boldsymbol{r_{1,1}} = \boldsymbol{S}$, $\boldsymbol{r_{2,2}} = \boldsymbol{V}$, and $\boldsymbol{r_{3,3}} = \boldsymbol{O}$.

(31)

0.	***S***	***V***	***O***
1.	***S***	*V*	*O*
2.	*S*	***V***	*O*
3.	*S*	*V*	***O***

Based on the equivalence discussed above, it is easy to see that pattern 0 is equivalent to the structure diagrammed in Figure A.3, which is, as Figure A.4 shows, composition **123** out of **1**23, 1**2**3, and 12**3**, provided that $S = 1$, $V = 2$, and $O = 3$.

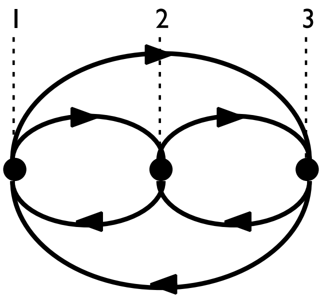


Figure A.3

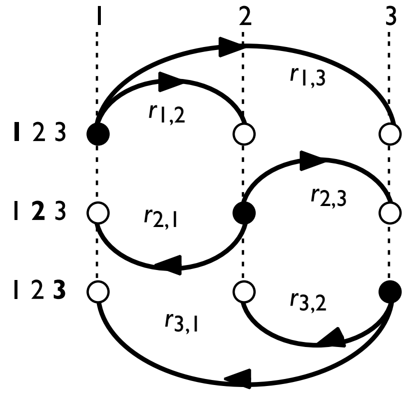


Figure A.4

Therefore, pattern composition/decomposition can be captured also in terms of diagram.

A.4.6 How words glue with each other

Glues such as *S*, *V*, *O* are not "pure" markers of grammatical roles such as subject, verb, and object. Rather, they are mnemonics of co-occurrence restrictions, different from one word to another. This is so even though skeletal form *S V O*, for example, is shared by most verbs, e.g., *S* ***admire*** *O*, *S* ***disgust*** *O*, *S* ***undergo*** *O*. Here, *S* and *O* are "specialized" for each of such verbs, and their contents are remarkably different. It is impossible to state in terms of the co-occurrence matrix what contents they have. I believe I am entitled to disclaim such a responsibility. It is **lexicography** that should be responsible for it, and I claim that pattern matching provides a well articulated candidate for the description of the **interface between lexicography and syntactic analysis**.

Another important implication follow from this. Words, if conceived of as schemas, have potentials of their own to combine with other units. Differently put, all such basic units of syntax are part of a large network of words. Specification of syntactic structure is, in a crucial sense, merely a selection of "nodes" in the network. For this, I admit that my conception is influenced by ideas of Hudson's word grammar approach, who (1998: 6) remarks as follows:

> ... In short, knowledge is held in memory as an associative network. What is more controversial is that, according to [word grammar], the same is true of our knowledge of words, so the sub-network responsible for words is just part of the total 'vast set of associations'.

Though Hudson does not mention connectionism, a conceptual link is obvious, since his point in note is that, given human memory is a vast set of associations, "[o]ur knowledge of words is our language, so our language is a network of associations which is closely integrated with rest of our knowledge" (1998: 6). It is reasonable to assume that such knowledge is reflected on semantic "contents" of glues.

In this connectionist system, moreover, "[p]hrases are," Hudson (1998: 2) explains, "implicit in the dependencies, but play no part in the grammar." This also forms a reason to dispense, at least conceptually, with an independent component to generate phrase markers like (15)a = $[_A [_B \ldots][_C [_D \ldots][_E [_F \ldots][_G \ldots]]]]$, where, under usual interpretation, A = S (or V^2), B = NP, C = VP (or V^1), D = V°, E = NP, F = AP, G = N (or N^1). This idea will be elaborated in subsequent discussions.

A.5 Role of Overlaps among Subpatterns

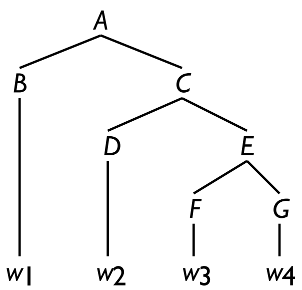
It is now time to see how composition of patterns, if defined as superposition, do without phrase structure. The key concept is **overlapping among subpatterns**. This is because composition is defined as superposition of subpatterns: without overlapping of patterns, no superposition is carried out. In fact, utilization of overlaps among units is the most straightforward and effective way to derive absolute ordering among strings.

A.5.1 What are to be superposed?

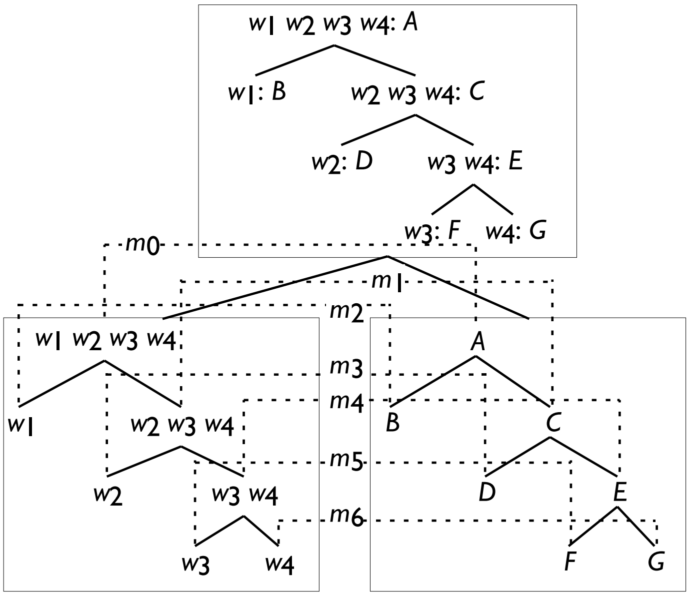
A note will be helpful. Since (32) is a lexical realization of (15)a, it is possible to reinterpret (32), which is assumed to encode the syntactic structure of *Bill undergoes an operation* [= (10)], in terms of "tree superposition", illustrated in (15)b′ instead of b.

(32) $[_{IP}$ *Bill* $[_{I'} [_I$ *undergoes* $][_{NP} [_{AP}$ *an* $][_{N'}$ *operation* $]]]]$

(15) a. $[_A [_B w_1][_C [_D w_2][_E [_F w_3][_G w_4]]]]$
b.



b′.



Tree superposition assumed here necessitates "node-by-node" matching conditions (indicated by links labeled m_0, ..., m_6) between tree structures, left and right at bottom.[10]

No matter what sense this kind of reinterpretation appears to make, I find it is circuitous. To provide evidence, let me discuss a simple example.

A.5.2 Conditions for the emergence of surface patterns

Suppose form $F = w_1\ w_2\ w_3\ w_4\ w_5$, where w_i is the i^{th} word of F. This matches terminal nodes of phrase marker $M = C_1\ C_2\ C_3\ C_4\ C_5$, where C_i is the i^{th} unit of a string of M's preterminal nodes (linear arrangement is assumed). Illustrated below is the first case where five "decontextuated" subpatterns, $w_{1.1}$, $w_{2.2}$, ... , $w_{5.5}$, are "inserted" into, or attached to, the "terminal" nodes of M (= 0).

(33)

0.	C_1	C_2	C_3	C_4	C_5
1.	$w_{1.1}$				
2.		$w_{2.2}$			
3.			$w_{3.3}$		
4.				$w_{4.4}$	
5.					$w_{5.5}$

Note that 0 must be generated "independently" by, for example, the base component of generative grammar.

Compare this with other cases where overlapping plays a role. First is a case where 4 subpatterns, $w_{1.1}\ w_{1.2}$, $w_{2.2}\ w_{2.3}$, ..., $w_{4.4}\ w_{5.5}$, overlapping each with other at length of 1, are superimposed into a "complete" pattern (= 0) at length of 5.

$$
(34)\quad
\begin{array}{llllll}
0. & w_1 & w_2 & w_3 & w_4 & w_5 \\
1. & w_{1.1} & w_{1.2} & & & \\
2. & & w_{2.2} & w_{2.3} & & \\
3. & & & w_{3.3} & w_{3.4} & \\
4. & & & & w_{4.4} & w_{4.5}
\end{array}
$$

Pattern 0 is obtained by vertically unifying $w_{i.k}$ and $w_{j.k}$ to eliminate prefix indices i and j.

Note that, in this and other cases below, pattern 0 need not be independently defined (by base rules, for example), as far as subpatterns 1, 2, ..., 5 are already defined. This is because pattern 0 is composed by superposing overlapping subpatterns.

If 3 subpatterns, $w_{1.1}\, w_{1.2}\, w_{1.3}$, ... , $w_{3.3}\, w_{3.4}\, w_{3.5}$, are superimposed into a complete pattern (= 0) at length of 5, then the following is given:

$$
(35)\quad
\begin{array}{llllll}
0. & w_1 & w_2 & w_3 & w_4 & w_5 \\
1. & w_{1.1} & w_{1.2} & w_{1.3} & & \\
2. & & w_{2.2} & w_{2.3} & w_{2.4} & \\
3. & & & w_{3.3} & w_{3.4} & w_{3.5}
\end{array}
$$

Note that in this case too, pattern 0 need not be defined independently.

If 2 subpatterns, $w_{1.1}\, w_{1.2}\, w_{1.3}\, w_{1.4}$, ... , $w_{2.2}\, w_{2.3}\, w_{2.4}\, w_{2.5}$, overlapping each with other at length of 3, are superimposed into a complete pattern (= 0) at length of 5, then the following is given:

$$
(36)\quad
\begin{array}{llllll}
0. & w_1 & w_2 & w_3 & w_4 & w_5 \\
1. & w_{1.1} & w_{1.2} & w_{1.3} & w_{1.4} & \\
2. & & w_{2.2} & w_{2.3} & w_{2.4} & w_{2.5}
\end{array}
$$

Finally, if only one subpattern, $w_{1.1}\, w_{1.2}\, w_{1.3}\, w_{1.4}\, w_{1.5}$ (overlapping with itself at length of 4?) is "vacuously" superimposed into a complete pattern (= 0) at length of 5, then the following is given:

$$
(37)\quad
\begin{array}{llllll}
0. & w_1 & w_2 & w_3 & w_4 & w_5 \\
1. & w_{1.1} & w_{1.2} & w_{1.3} & w_{1.4} & w_{1.5}
\end{array}
$$

It deserves a note that the minimum subpattern length for there to be overlaps is two, and conversely, the maximum (in this case) is five, since there is no subpattern.

Note also that (33), where subpatterns have no overlaps, and (37), where one and only one subpattern equals the whole, are two special cases in opposite directions. In both cases, the notion of subpattern makes no sense. In (33), subpatterns are all context-free units. Such subpatterns, with no overlap, are impotent to combine with each other by themselves, and for this reason, pattern 0 is necessary as a reference pattern to combine them up to a complete pattern. In (37), part and whole are the same.

It is easily observed that the length (L) of each subpattern and the number (N) of subpatterns required to constitute a whole are in relation $N + L > \max(N)$. Confirm this by seeing $(N{:}L) = (1{:}5)$ in (33), (2:4) in (34), (3:3) in (35), (4:2) in (36), and (5:1) in (37).

I claimed earlier that a pattern emerges as subpatterns interact with each other, even in the simplest ways. My claim is supported by what (34), (35), and (36) show concisely in light of how pattern 0 emerges out of the interaction of subpatterns. There are two conditions that must be observed, specified as follows:

(38) **Condition I.** Some, if not all, subpatterns are properly larger than the minimum size (of lexical units).
Condition II. Some, if not all, subpatterns are properly smaller than the maximum size (of the whole pattern).

Interestingly enough, I believe, these conditions contradict, in a sense, the well established assumption that linguistic form consists of a combination of ultimate units, since they conceptually blur the explanatory role that ultimate units are expected to play.[11]

Clearly, ultimate units, no matter how they are to be defined, can do nothing syntactically interesting. This, it seems, is the reason why syntax in a broader sense must exist. To elaborate this idea, it will be useful to see how the one and only string "12345" is uniquely determined by specifying such sets as given in (39), where $S[i \leqslant l \leqslant j]$ indicates a set of substrings at length of l, with the shortest is of length i and the longest is of length j.

(39) i. $S[2 \leqslant l \leqslant 2] = \{12, 23, 34, 45\}$
ii. $S[3 \leqslant l \leqslant 3] = \{123, 234, 345\}$,
$S[2 \leqslant l \leqslant 3] = \{123, 23, 345\}$, etc.
iii. $S[4 \leqslant l \leqslant 4] = \{1234, 2345\}$,
$S[3 \leqslant l \leqslant 4] = \{1234, 345\}, \{123, 2345\}$, etc
$S[2 \leqslant l \leqslant 4] = \{1234, 45\}, \{12, 2345\}$, etc.

Note that no pattern can be composed uniquely without overlap. Thus, (39)i is impotent to determine 12345.

(40) $S[1 \leqslant l \leqslant 1] = \{1, 2, 3, 4, 5\}$

This case illustrates the way most linguists view the lexicon. I will return to this issue later.

A.5.3 Efficiency-motivated pervasiveness of triplets

Under these conditions, I can discuss the emergence of a recurrent pattern *X–R–Y*, with *X* being interpreted as "subject", "specifier", and the like, *R* as "verb", "head" and the like, and *Z* as "object", "complement" and the like. I will never claim, without insight into emergence, that so-called specifier-head-complement scheme is a characteristics of UG. Rather, it is better to state that triplets of the form *X–R–Y* are merely a special case of *n*-ary dependencies, but they are "optimal" in that neural equilibrium. Whether one can or cannot say truthfully that this specifies or manifests part of an "innate knowledge of language" is another problem. With Elman, *et al.* (1996), I will resist to such a simplism.

A.5.4 What is the lexicon, and where is it?

It is quite interesting to note that the set of subpatterns in (40), which are ultimate units rather than substrings, is equivalent to the standard conception of what is called "the lexicon", in which lexical items are listed without being related to each other. But as is already clear, it makes a great difference whether (39) or (40) are conceived of as a fragmentary description of the lexicon.

Implicit in formulation (40) is a **view of the lexicon as an appendix to grammar**, which necessitates a system of rules to independently define patterns to be completed (12345 in this case), irrespective of whether they are phrase structures or not. So, the difference of (39) from (40) is more drastic than it seems.

In addition, sets of the form $S[i \leqslant l \leqslant 5]$ are all "vacuous" to compose "12345", since they already contain the pattern "12345", which is to be composed:

(39) i. $S[5 \leqslant l \leqslant 5] = \{12345\}$,
ii. $S[4 \leqslant l \leqslant 5] = \{1234, 12345\}, \{2345, 12345\}$, etc.
iii. $S[3 \leqslant l \leqslant 5] = \{1234, 12345\}, \{2345, 12345\}$, etc.
iv. $S[2 \leqslant l \leqslant 5] = \{1234, 12345\}, \{2345, 12345\}$, etc.

This does not imply, however, that such sets, if identified as partial descriptions of the entire lexicon, are not real. This assertion will become more convincing if we take it into consideration that the lexicon may be a **self-organizing system**.

Furthermore, this forms a basis for an answer to the question of whether pattern composition is ordered or not. If the answer is yes, there is derivation, and if not, there is none. There exists a few classes of fact (e.g., quantification) that seem to be sensitive to the order of composition. I have touched on one or two instances of this in Chapter 6.

Generally, however, it varies from one situation to another where composition is ordered intrinsically or extrinsically. Ordering of composition, if anything, only leads to certain "side effects" such as "scope ambiguity" in quantification. It is thus compatible with the claim that composition of subpatterns is freely ordered.

Constraints on pattern composition should be stated in certain ways, but, on the same connectionist ground as above, they need not be stated as derivations. They must be simultaneous satisfaction of combinatoric constraints in the sense of Lakoff (1993), in which I found a hint for conceiving of grammars that work "inductively" rather than "deductively", or in the manner of Post production system.

A.6 Scale Effects in Syntax

I have to acknowledge that linguistic units such as NP, VP, PP, in short **phrasal units**, could be treated improperly in the proposed framework. I devote discussions in this section to this issue, noting that PMA account of phrasal units is still under development.

A.6.1 Recognition of phrasal units

To see the sort of problems mentioned above, let us examine a few examples.

For *Bill undergoes an operation* [= (10)], there are as many parses as possible. First, the following gives the "vacuous" parse of (10).

(41) ***<Bill undergoes an operation>***

In other words, vacuous parse results in no segmentation of a surface formation, thereby taking the whole as the single part. Interestingly, so-called *S* (for **sentence**) or IP (for Inflection(al) Phrase) is exactly this kind of object.

Next to this is a parse into two parts. In this case, the following two ones can be specified:

(42) i. ***<Bill undergoes*, *an operation>***
ii. ***<Bill*, *undergoes an operation>***

For obvious reasons, I take a parse to an ordered set.

Next to this is a parse into three parts.

(43) ***<Bill*, *undergoes*, *an operation>***

This gives the following:

(44)	0.	***Bill***	***undergoes***	***an operation***
	1.	***Bill***	*V*	*(O)*
	2.	*S*	***undergoes***	*O*
	3.	*S*	*V*	***an operation***

Note first that it is better to say that the most basic parse of *Bill undergoes an operation* is (45) rather than complex (19).

(45)	0.	***Bill***	***undergoes***	***an operation***
	1.	***Bill***	*V*	*(O)*
	2.	*S*	***undergoes***	*O*
	3.	*S*	*V*	***an operation***

This is basic because the parse makes reference to only *S V O*.

The two analyses in (19) and (45) are at least apparently incompatible. The reason is that while *an operation* as a whole match O in (45)2, it has two subpatterns *an* and *operation* in (19)2, only the latter of which matches O. To encode this explicitly, subindexing convention is introduced to the following effect:

(46)	0.	***Bill***	***undergoes***	***an***	***operation***
	1.	***Bill***	*V*	*(O_1*	*O_2)*
	2.	*S*	***undergoes***	O_1	O_2
	3.1	*S*	*V*	***an***	*N*
	3.2	*S*	*V*	*(D)*	***operation***

Subpatterns are segmented. O_1 and O_2 are the first and second segments of *O*. Subpatterns 3.1 and 3.2 are the first and second segment of (19)3. By subindexing convention, a C/D table is made able to encode such intermediate, **phrasal units** as *an operation* (= NP), and *undergo(es) an operation* (= VP), if any.

Another kind of relation must be assumed, of course, to accommodate the difference in *O*-matching in (19) and (45). Note that *an operation* is a noun (phrase) because *operation*, its head, is a noun. My concern here is guaranteeing that composition of 3.1 and 3.2 results in no change of category. This is exactly the problem of **head**. For this, I simply assume that *an operation* *O*-matches 2 (and 1) because its head *operation* is *O*.

A note is in order. If composite units like 3×4 as NP and $2\times3\times4$ as VP are legitimate in (47), then it should be naturally questioned that units like 1×2 and even $1\times2\times3$ in (48) are not legitimate.

(47) 1,2.	***Bill***	***under-***	***go-***	***es***		*O*
3,4.	*S*		*V*		***an***	***operation***

(48) 2.	*S*	***under-***	***go-***	***es***		*O*
1,3,4.	***Bill***		*V*		***an***	***operation***

PMA indeed assumes that units like $1\times2\times3$ = ***Bill V an operation*** are implicit units of *Bill undergoes an operation*, if they are not phrasal. Aware that this is a rather controversial claim, I gave arguments in Chapter 6.

A.6.2 Morphological statements scattered in syntax

The composition table in (19) can be revised to reflect "morphological" statements, together with "syntactic" statements on larger scales. Illustrated below is a morphologically detailed analysis of (19) in which pattern 2 is replaced by morphological statements 2.1-2.3, where *under*-, *go*, and *-es* are combined into a single (prosodic) word, *undergo*.

(49)							
	0.	***Bill***	***under***	***-go***	***-es***	***an***	***operation***
	1.	***Bill***		*V*			*(O)*
	2.1	*S*	***under***				*O*
	2.2	*S*	*P*	***-go***			
	2.3	*S*		*V*	***-es***		
	3.					***an***	*N*
	4.	*S*	*P*			*(D)*	***operation***

Another notational convention is assumed here. In statements of the form, "… *x* … *-y* …" and "… *y*- … *x* …", symbol "-" stands for **adjacency marker** which indicates that ***y***, an affix, combines with its target *x* to form ***xy***, and ***yx***, lexically or prosodically, though units other than *x* may serve as arguments of ***y***.

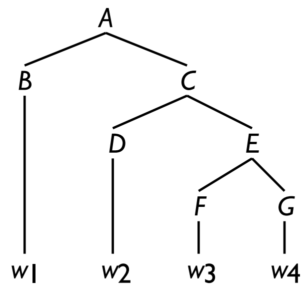
I will discuss how PMA deals with morphological properties in more detail in Section A.7.

A.6.3 Getting sequences of derivation out of analysis

Under the assumptions made so far, it is clear that so-called tree diagrams, or more exactly phrase-markers, like (15)a, b, repeated here for convenience, can be dispensed simply because there is a better means to detail properties of syntactic structure. It is co-occurrence matrices like (50), which schematizes specifications in (19) above, on the condition that nodes *B*, *D*, *F*, and *G* in (15)a, b correspond to *S*, *V*, *A*, and *O* in (50), whereas "phrasal nodes" *A*, *C*, and *D* in (15)a, b have no counterpart in (50).

(15) a. $[_A\, [_B\, w_1\,][_C\, [_D\, w_2\,][_E\, [_F\, w_3\,][_G\, w_4\,]]]$

b.



(50)					
	0.	w_1	w_2	w_3	w_4
	1.	w_1	V	$(O_1$	$O_2)$
	2.	S	w_2	O_1	O_2
	3.	S	V	w_3	N
	4.	S	V	(D)	w_4

I will not discuss in detail here why phrasal nodes in phrase markers have no counterparts in co-occurrence matrices, only noting the likelihood of phrasal nodes being an artifact of hierarchical phrase structure. One hint is that, as I will discuss in more detail, the co-occurrence matrix here represents an abstract structure diagrammed as follows:

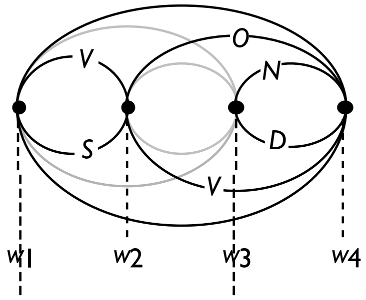


Figure A.5

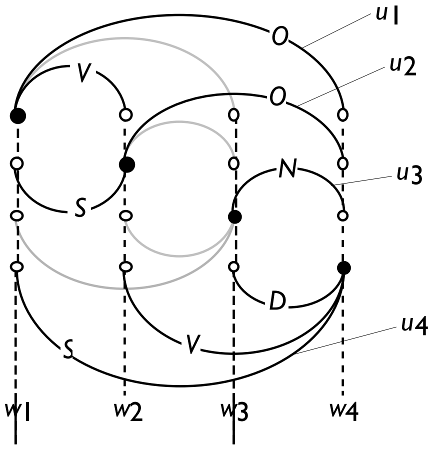


Figure A.6

The diagram on the left is the composition of the right, where solid links correspond to "labeled" binary relations in (50) (except $r_{i,i}$), and dimmed links to unlabeled relations.

If phrasal nodes *A*, *C*, and *E* in the phrase marker tree are also (implicitly) encoded in this diagram, they are nothing but "loops" or "circuits" such as:

(51) $E = w_4 \neq w_3 \neq w_4$
$D = w_2 \neq E \neq w_2\ (= w_2 \neq w_4 \neq w_3 \neq w_4 \neq w_2)$

$$A= w_1 \neq D \neq w_1 \ (= w_1 \neq w_2 \neq w_4 \neq w_3 \neq w_4 \neq w_2 \neq w_1)$$

Without proof, I note here that constituency in general is encoded by loops of this sort, and that all kinds of restrictions that make reference to phrasal constituents like NP, VP, and S, can be reinterpreted as restrictions on formation of such loops.

I believe the discussion provided here is convincing enough to allow me to claim for the descriptive superiority of co-occurrence matrices over phrase markers, on the ground that all the possible relations among units are implicitly or explicitly encoded by co-occurrence matrices. However, to be more faithful, I also suggest readers who find this unconvincing to consult Hudson's works (1976, 1980[a, b], 1981, 1984) for other kinds of arguments for similar claim.

A.7 Morphology as Integrated into Syntax (Rather than Separated from it)

Now turn to morphological details of the analysis of (10), repeated here for convenience.

(10) *Bill undergoes an operation.*

As I have discussed, the syntactic analysis of this form is given as the following C/D table, repeated here for convenience:

(19)	0.	***Bill***	***undergoes***	***an***	***operation***
		›	›	›	›
	1.	***Bill***	V_1	$(O_{1.1}$	$O_{1.2})$
	2.	S_2	***undergoes***	$O_{2.1}$	$O_{2.2}$
	3.	S_3	V_3	***an***	N_3
	4.	S_4	V_4	(D_4)	***operation***

The analysis given in (19) would be sufficient for usual purposes, but it is clearly too rough for a morphological analysis.

It is assumed that syntactic analysis should be **scale-sensitive** in that it should be done relative to an appropriately determined size of units. It is better, however, if one can freely shift from one scale to another, rather than confining one's analysis to a predetermined scale, e.g., of prosodic words, lexical words, morphemes. Pattern matching analysis is superior, I claim, in that it provides the flexibility desired.

Illustrated below is another analysis of (10), a slightly detailed version of (19), in which subpatterns 2.1 and 2.2 encode morphological statements that *undergo* and *-es* are combined into a "lexical" word *undergo*, as distinguished from a "prosodic" word, *undergoes*.

(52)	0.	***Bill***	***undergo***	*-es*	*an*	***operation***
	1.	***Bill***		*V*	(O_1	O_2)
	2.1	*S*	***undergo***		O_1	O_2
	2.2	*S*	***V***	*-es*		
	3.	*S*	*V*		*an*	*N*
	4.	*S*	*V*		*(D)*	***operation***

In this analysis, my analysis stops at the scale of lexical words. But this is not a necessary decision. There are smaller scales. Illustrated below is another analysis, where a "complex" (compound-like) lexical word *undergo* is further analyzed into "simple" lexical words *under* and *go*.

(49)	0.	***Bill***	***under-***	***go***	*-es*	*an*	***operation***
	1.	***Bill***		*V*		(O_1	O_2)
	2.1.1	*S*	***under-***	*V*		O_1	O_2
	2.1.2	*S*	*U?*	***go***			
	2.2	*S*		*V*	*-es*		
	3.	*S*	*P*			*an*	*N*
	4.	*S*	*P*			*(D)*	***operation***

There is another kind of flexibility that can be benefited from.

A.7.1 Functional composition

The co-occurrence matrix in (52) claims that *an operation* is *O* of *undergo(es)* if and only if it is *O* of *under* rather than of *go*. For one thing, *go* is unable to take nouns like *operation* as its object. For another, the relation of *under* and *go* is an instance of **functional composition**.

A.7.2 Notion of *scattered* morphology

Pattern matching analysis conceives of morphology as something **scattered** around. Morphology is scattered because it is by no means necessitated that analysis is confined to a "consistent level", e.g., of derivational morphology, as distinguished from inflectional morphology. More specifically, analysis of *undergo* into *under-* and *go* by no means necessitates analysis *operation* of *operat(e)* and *-ion*, as the following illustrates:

(53)	0.	***Bill***	***under-***	***go***	***-es***	***an***	***operat***	***-ion***
	1.	***Bill***		*V*		*(*O_1	$O_{2.1}$	$O_{2.2}$*)*
	2.1.1	*S*	***under-***	*V*		O_1	$O_{2.1}$	$O_{2.2}$
	2.1.2	*S*	*U?*	***go***				
	2.2	*S*		*V*	***-es***			
	3.	*S*	*P*			***an***	N_1	N_2
	4.0.1						***operat***	
	4.0.2	*S*	*P*			*(D)*	*S V O*	***-ion***

Segments of *O* in 1 and 2.1.1, and likewise segments of *N* in 3 are better interpreted as "feature bundles" that *an*, *operat(e)*, and *-ion* correspond to. Note also that *S P* must disappear from 4.0.1.

Note that there is an obvious correlation between linguistic levels and indices of the form *i.j.k* (e.g., 2.1.2). Prefix *i* indexes a scale of analysis in such a way that *i* indexes the scale of (prosodic) words, *j* the scale of complex words, and *k* the level of simple words, or possibly morphemes. So, it is possible to replace the analysis in by the following, if really in need.

(54)	0.	***Bill***	***under-***	***go***	***-es***	***an***	***operat***	***-ion***
	1.0.0	***Bill***		*V*		*(*O_1	$O_{2.1}$	$O_{2.2}$*)*
	2.1.1	*S*	***under-***	*V*		O_1	$O_{2.1}$	$O_{2.2}$
	2.1.2	*S*	*U?*	***go***				
	2.2.0	*S*		*V*	***-es***			
	3.0.0	*S*	*P*			***an***	N_1	N_2
	4.0.1						***operat***	
	4.0.2	*S*	*P*			*(D)*	*S V O*	***-ion***

This kind of consistency, as far as I can see, makes little sense. It may be conceptually necessary but it is by no means practically or even factually necessary. By this, I suggest that the notion of linguistic level is not a coherent notion, if not a self-contradiction, since there should be, and indeed are, as many levels as the degrees of complexity of units.

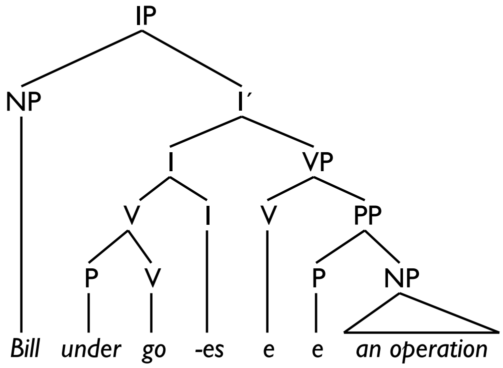
It is already clear, I think, that pattern matching analysis succeeds in integrating morphological issues into syntax, rather than segregating it from syntax. On this view, morphology is merely word-internal syntax, irrespective of how "words" are defined. Therefore, it is no surprising even if affixes like ***under****go* have "argument structure" of their own, considering the fact that they are relational in nature.

A.7.3 Pattern matching analysis in contrast with head-movement analysis

If attempted, head-movement analysis is very likely to be proposed for (10).

(55) a. [${}_{\mathrm{IP}}$ [${}_{\mathrm{NP}}$ *Bill*][${}_{\mathrm{I'}}$ [${}_{\mathrm{I}}$ [${}_{\mathrm{V}}$ [${}_{\mathrm{P}}$ *under*]${}_{j}$-[${}_{\mathrm{V}}$ *go*]]${}_{i}$-*es*][${}_{\mathrm{VP}}$ [${}_{\mathrm{V}}$ e_i][${}_{\mathrm{PP}}$ [${}_{\mathrm{P}}$ e_j][${}_{\mathrm{NP}}$ *an operation*]]]]

b.



Such authors as Baker (1988) indeed treat incorporation roughly in this fashion.

Such treatment, in my view, brings about an unnecessary complication of a simpler fact. Notably, the intended effect of a such syntactic movement is captured more naturally by **glue sharing** in co-occurrence matrices. The syntactic property of *undergo(es)* is accounted for in terms of *S*-sharing between *S* ***under***-*V O* and *S* ***go*** assuming *V* to match ***go***.

To illustrate relevant points, the syntactic structure encoded by (49) can be diagrammed in the following way:

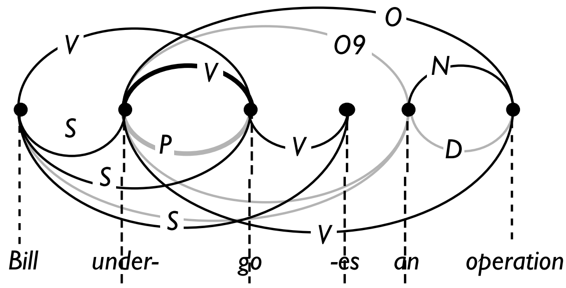


Figure A.7

This diagram shows the interesting property of ***under****go* that I mentioned above. Here, *go* is a sort of object of prefixal preposition *under-* which takes three arguments, *S* = *Bill*, *V* = *go*, and *O* = *operation*. In this respect, one may take *under-* to be a special kind of auxiliary verb relative to *go*.

To take another perspective, *go* in this construction is a sort of "light verb" which gives tense-bearing ability to *S* ***under*** *O*, which lacks it. In this respect, *S* ***under*** *O* is a parasite of *S* ***go***.

My point is that these interesting properties are all missing, or at least obscured, in tree structures like (55)a, b.

A.8 Subpatterns Are More than Subcategorization Frames

It is not pointless to remark that the encoding scheme utilized for u_1 = ***Bill*** *V*, u_2 = *S*

undergoes O, u_3 = ***an*** N, u_4 = *V* ***operation*** has far from a superficial resemblance to so-called **subcategorization frames**. For example, it is usually assumed in generative linguists that *undergo* is listed “in the lexicon” as a lexical entry of the form:

(56) <*undergo*, V, $[_{VP}$ __ NP]>

To annotate, it states *undergo* is a verb (encoded by V) such that it combines with a NP to form a node VP.

Or in greater detail, the entry may be:

(57) < **phonology** = /ˌʌndər'gow/;
syntax = V, $[_S$ N_1P $[_{VP}$ __ N_2P]], where N_1 is [+human], and N_2 is [+abstract], etc;
semantics = $\mathscr{D}(N_1)$ ENDURES $\mathscr{D}(N_2)$ ... >

$\mathscr{D}(x)$ encodes *x*’s denotation.

In a sense, it is not pointless to think that the notion of syntactic pattern is a conceptual revision on the notion of subcategorization frame. In fact, it will be unfaithful if recent conceptual revisions in generative theorizing about the relevant issues are ignored, in favor of “cognitive” approaches. To quote from Chomsky (1986*b*: 86),

> Having virtually eliminated phrase structure rules through recourse to certain general principles and properties of the lexicon, we now consider just what information the latter must contain. In the first place, the lexicon presents, for each lexical item, its (abstract) phonological form and whatever semantic properties associated with it. Among these will be ‘‘selectional properties’’ of heads of constructions: nouns, verbs, adjectives, and particles (prepositions and postpositions, depending on how the head-complement parameters are set in the language). [...] Is it also necessary to specify in the lexicon properties of categorial selection (c-selection), for example, that hit takes an NP complement (*hit John*)? The latter specification seems redundant; if hit s-selects a patient, then this element will be an NP [according to Canonical Structural Realization]. If c-selection is redundant, in general, then the lexicon can be restricted to s-selection.

While sharing certain basic insights, I take, however, a more radical position. I take the lexicon to contain syntactic information, as well as categorial information. This eliminates a component called “base”, thereby attaining a version of what Diehl (1981) claims is the “most restrictive theory of grammar”.

This does not mean that patterns are a “notational variant” of, and therefore reducible to, subcategorization frames.[12] There are two major differences, among others, to which I now turn.

A.8.1 Argument 1

First, glues *S*, *V*, *O*, used in patterns are not "meaning-free" syntactic categories NP, V, although certain symbols (e.g., *V*, *P*) are shared. Admittedly, the schematization proposed above indeed masks some complication. This can be revealed by comparing ***Bill*** *undergoes ...* (as subject of *undergoes ...*) with *Ann saw* ***Bill***, *S* ***undergoes*** *O*, with *the subject* ***undergoes*** *Raising*, and ... *undergoes ...* ***operation*** with ***operation*** *is ahead of us*, as the following contrasts encode:

(58) a. ***Bill*** *undergoes an operation.* [= (10)]
a′. *Ann saw* ***Bill***.
b. *Bill* ***undergoes*** *an operation.*
b′. *In this sentence, the subject* ***undergoes*** *Raising.*
c. *Bill undergoes* ***an operation***.
c′. ***Hardship*** *is ahead of them.*

Two different uses of *Bill* are contrasted in (58): (58)a illustrates *Bill* as subject and (58)a′ *Bill* as object. Similarly, two uses of *undergoes*, and two uses of *operation* are contrasted in the b-b′ and c-c′ pairs.

Those contrasts manifest **context-sensitive identification of grammatical units**, based on which PMA claims that:

(59) All uses of a word must be encoded differently if syntactic contexts in which the word participate are different.

This means that there are as many lexical encodings for *Bill*, for example, as there are different syntactic contexts for it. So, it is claimed further:

(60) i. **Syntactic contexts for words are as important as words themselves**, or more explicitly,
ii. **Syntactic contexts are a legitimate component of words**
iii. **Knowledge of such contexts constitutes grammar**

This claim sharply differentiates my approach from others.

This is an important point that motivates the entire framework of pattern matching analysis, and I will discuss it more thoroughly in Appendix B by discussing Jane Hill's **classification through word use** model (Hill 1982, 1983, 1984; Arbib, Hill, and Conklin 1987) and Elman's results (Elman 1990, *et seq.*) from connectionist simulations, especially of the necessity of "starting small" (Elman 1993; Elman, *et al.* 1996). With these notes, let me anticipate some crucial points here.

A.8.2 Need for well-constrained context-sensitivity

Admittedly, fully context-sensitive encoding may seem too redundant, and at first glance, such a claim is too absurd to accept, especially for generativists who love "generality". But what is really in need is "optimal" redundancy, and there is indeed a need for well-constrained context-sensitivity.

I claim, with Rumelhart and McClelland (1986) and other "connectionists", that such utilization of redundancy is at the heart of the mechanism that underlines the human mind, namely **distributed representation and control**, though the former is usually more emphasized for unclear reasons.

For more clarity, it is helpful to note that patterns like ***Bill** V (O)*, *S **undergoes** O*, *S V **an operation*** are **schemas** not only in the sense popular in cognitive linguistics, but also in the sense used in Arbib, *et al* (1987), Arbib (1989).

Adopting the latter view, I suppose that words are schemas that are "acquired" through exposition to a mountain of uses only a tip of iceberg of which are given as follows:

(61) a. ***Bill** met Ann.*
a′. ***Bill** met Ann frequently.*
b. ***Bill** ate the pizza.*
b′. ***Bill** ate pizza willingly.*

(62) a. *She **undergoes** misery.*
a′. *She **undergoes** intolerable misery.*
b. *The subject **undergoes** Raising.*[13]
b′. *The subject **undergoes** Raising of Subject to Object.*

(63) a. *Ellen knows no **operation**.*
a′. *Ellen knows no **operation** in youth.*
b. *There is no **operation**.*
b′. *There is no **operation** in front of you.*

To be fair, I note that the idea of patterns as schemas is a conceptual successor of **wickelphonology** that Rumelhart and McClelland (1986), after Wickelgren (1969), used to provide a connectionist account of past-tense formation, on the one hand, and a companion of Langacker's **usage-based** model of grammar (1987, 1991*a*,*b*), on the other. The notion of patterns is inherited from that of wickelphones: the idea of context-relative encoding of units.

Relating to my point, the adoption by Rumelhart and McClelland (1986) of wickelphones (coupled with "wickelfeatures" for distributed representation for them) was one of the most severely attacked parts. Rumelhart and McClelland's argument for a connectionist account of grammatical performance encountered a lot of criticisms. Two of the most notable are: Pinker and Prince (1988) and Fodor and Phylyshyn (1988). So, it is not surprising that utilization of such context-sensitive units as wickelphones was avoided in later connectionist research.

As I discuss in Appendix B, postulation of schematization procedure allows the notion of patterns to endure most of the criticisms exercised against the notion of "wickelphones".

A.8.3 Argument 2

Return to the original problem of whether patterns in the assumed sense are or are not a notational variant of subcategorization frames.

In addition to the first reason given, there is another, more crucial reason. Patterns, as conceived here, know no limit of environmental specification, at least theoretically, while subcategorization frames, for theory-internal reasons, should be "localized" within a limited kind of environment, e.g., VP (and possibly S) to meet the notion of "dominance". For example, the syntactic portion of subcategorization frame of *undergo*, as given above, should be characterized as:

(64) $[_{VP}$ __ NP]

rather than:

(65) a. $[_{S}$ NP __ NP], or
b. $[_{S}$ NP $[_{VP}$ __ NP]]

Labels such as S, VP are loosely chosen, in view of the fact that all the technical issue like whether or not S = V^2, VP = V^1 or whether or not S = CP are totally irrelevant here.

In most versions of generative grammar, there is an independent component, called the base, assumed to take care of phrase structure. It generates, by assumption, phrase markers like [NP [V NP]] independently of any lexical specifications. So, a ban on redundancy motivates why lexical items like *undergo*, as a verb, are kept from having a subject of their own.

This motivation makes sense within a theory that has to have base component for reasons that are far from well founded. But it is clearly possible, as Diehl (1981) claims correctly, to eliminate the base component altogether in favor of the lexical component, as far as so-called selectional restrictions have to be incorporated into grammar. He points out that all verbs to be inserted into preterminal V node must already "know" what their subject (and object, if any) is; otherwise, selectional restrictions could never be effective, nor stated. This means that the information about what noun serves as subject (and object) of what verb is redundantly specified, once configurationally in a phrase structure tree (X = Subject if and only if X is NP/$[_{S}$ __ $[_{VP}$ V NP]]), and once more selectionally somewhere in lexical specification, though there seems no consensus among researchers about the latter point. In generative grammar, this redundancy is arbitrarily resolved by assuming, I claim preposterously, that grammar has, in addition to the lexical component to specify selectional restrictions, another base component, which comprises, among others, such rules as a or b.

(66) a. S → NP VP, VP → V NP
b. $X'' \to \{Y'', X'\}$, $X' \to \{X, Z''\}$, given X = V, N, ...

But, as Diehl (1981) points out, it is not only possible but also empirically plausible to eliminate the base component in favor of the lexical component as far as grammar should exclude sentences of the following sort.

(67) a. **Colorless green ideas sleep furiously.*
b. **Sincerity admires John.*

Of course, it is possible to assume that sentences here are grammatical but they are merely unacceptable. We believe that expressions in (67) should be grammatical for consistency with stating that the following are grammatical and acceptable.

(68) a. *Revolutionary new ideas appear infrequently.*
b. *John admires sincerity.*

Stating that expressions in (67) are unacceptable but grammatical contradicts with what is accepted as a main contribution of generative grammar to linguistics. Note that if sentences in (67) are unacceptable but still grammatical, then it would be inconsistent to claim that sentences in (69) are ungrammatical, and **therefore** unacceptable.

(69) a. *Who_i did he_i say that $Sean_j$ dislikes t_i?*
b. *Who_i did his_i mother disgust t_i?*

In generative literature, sentences of this sort are explained to be "ungrammatical" for their respective strong and weak "crossovers" in the sense of Postal (1971).

Rarely admitted or even recognized, most discussions in generative linguistics are faced with a sort of deep dilemma over the unacceptability/ungrammaticality differentiation. If expressions in (69) are grammatical but unacceptable for lexical reasons, then crossover phenomenon has nothing to do with competence, and therefore it need not be accounted for in terms UG. If the expressions are ungrammatical, and therefore unacceptable, then it is impossible to distinguish expressions like (67), as well as ones in (69), from the following.

(70) a. ***Infrequently ideas green appear new.*
b. ***John sincerity admires.*

In most cases, it seems, generative linguists say that unacceptable sentences are ungrammatical, but, once inconsistency is revealed, they are always ready to convert their words and say that they are in fact grammatical but unacceptable for "unknown" reasons, thereby arbitrarily exchanging grammaticality with acceptability.

I will never say that the grammaticality/ungrammaticality distinction and the acceptability/unacceptability distinction are a "matter of degree", despite arguments of leading cognitive linguists such as Lakoff and Langacker. Plainly, such claim is irresponsible, and fails to account for a lot of facts of language syntax.

Discussion so far reveals that it is essentially obscure whether given deviant sentences are deviant due to structural factors or selectional factors or both kinds of factors. In any case, pattern matching analysis departs from such a quagmire of a situation by rejecting any attempt to reduce co-occurrence conditions for lexical units to the tree-based, hierarchically motivated notion of domination. PMA is an attempt to replace the notion of hierarchical structure, which phrase marker is utilized to represent, by the notion of compositional dependency on the dimension of precedence/postcedence.

A.8.4 Note on the correlation between precedence and dominance

Recent theories of grammar put more focus on dominance, or dependency, than order. This move is perhaps influenced by the distinction introduced by Generalized Phrase Structure Grammar (Gazdar, *et al.* 1985). Gazdar, *et al.* point out that the structure of a sentence consists of two independent relations: **immediate dominance** (ID) relation, on the one hand, and of **linear precedence** (LP) relation, on the other.

Gazdar, *et al.* (1985: 50, 248) claim that the following three linear precedence statements are sufficient to account for most of the linear arrangements.

(71) a. [SUBCAT] < ~[SUBCAT]
b. [+N] < PP < VP
c. [CONJ a_0] < [CONJ a_1] where a_0 is in {*both*, *either*, *neither*, NIL} and a_1 is in {*and*, *but*, *or*, *nor*}

Roughly, (71)a states that lexical heads (e.g., N, A, V, P), if any, precede their complements. On the other hand, (71)b states that PP follows any N^n (and A^n, n= 0, 1, 2); and VP follows any P^n (n= 0, 1, 2). (71)c is a special statement about the linearization in coordinate structure.

Statements of this sort work fall under specification of dominance. Thus (71)b, for example, has nothing to do with *Larry peeled oranges with a small knife*, where there is a PP that does not follow VP. Such an instance is licensed by the fact that it is a VP-adjunct.

Such treatment will not work in the proposed framework of pattern matching analysis. The reason is obvious. We do not rely on dominance by which phrasal units are defined. In fact, what I attempt to do by PMA is to make linear precedence absorb immediate dominance.

This program is conceptually encouraged by computational works on language learning such as Hill's (1982, 1983, 1984) and Elman's (1990, *et seq.*), which virtually suggest that so-called hierarchical (phrase) structure "emerges". Elman showed, for example, that simple recurrent networks can learn from the surface distribution of units the crucial statistics of co-occurrences, thereby finding and internalizing lexical categories such as nouns and verbs, and presumably functional/grammatical categories subject and object.[14] It seems that it is possible to seriously attempt to account for surface formations by generalizing from them alone, resisting the temptation to deriving them from some structures "hidden in the deep".

A.9 Concluding Remarks

Based on results and considerations I have discussed thus far, I am now able to conclude this chapter.

Pattern matching analysis is a promising approach to the syntax of natural language, which is based on methods of **pattern composition based on superposition** and **pattern decomposition based on diagonalization**, and a theory of **emergence of subpatterns through schematization of token-based decomposition**. I claim that PMA describes syntactic structures realistically, rather than formally, and provides good tools for the reliable analysis of natural language syntax.

I also claim that PMA can replace phrase structure anlaysis if it successfully handles all effects attributed to phrase structure analysis. For this, however, there remain several problems. One of them is the description of the effects of phrasal units. For this, PMA appeals to the notion of matching scale. Although it provides some insights into the interaction of syntax and morphology, I have to concede that most such effects are not properly described yet. Despite this, though, I believe that future revisions will resolve this problem and other related problems.

Notes

1. By *constructionist*, I intend a theory whose accounts rely on (**grammatical**) **constructions**, which are defined as form-meaning pairs. Fillmore, Kay, and O'Connor (1988), Goldberg (1995), Kay (1997), among others, exemplify a constructionist theory.

2. This means, in a somewhat indirect way, that the **grammar-lexicon distinction** is blurred, though not utterly lost. I will discuss this point in the last section.

3. For consistency, PMA assumes, though controversially, an optional occurrence of O in *Bill* = ***Bill*** *V (O)*. This unnaturalness can be resolved either by assuming that O in intransitives is a zero form (presumably a reflexive pronoun) which is blocked from phonetical realization; or,

alternatively by assuming two subpatterns, namely *Bill* = ***Bill*** *V* and *Bill* = ***Bill*** *V* O. Without good evidence or a good argument, I have arbitrarily decided on the former option.

4. PMA does not posit a morphophonological rule to convert *a* to *an*, or vice versa. Rather, it makes use of **underspecification** such that *a(n)* describes {*a*, *an*}.

5. For simplification, I ignore here the fact that *operation* = *S V (D)(AdN)* ***operation*** is itself a composition of two subpatterns, i.e., *operation* = *S V (D)* ***operation*** and *operation* = *(D) (AdN)* ***operation***. The former encodes *operation* as O of *V*, whereas the latter encodes it as an argument of *AdN*.

6. For consistency, I assume notation ***uwv*** instead of Rumelhart and McClelland's subscript notation $_{u}w_{v}$.

7. It should be noted that /k/ and /t/ are respectively special cases of /...k/ and /t.../. This means that /kæt/ is /...kæt.../.

8. Wickelgren's original version was criticized more earlier by Halwes and Jenkins (1971) for the same reasons.

9. A subtle question may arise of whether or not those word as schemas take the form of trees. The question is largely open, but I arbitrarily assume it is not, simply because it is unmotivated.

10. Interestingly enough, the tree diagram at top is analogous to what is called an "analysis tree" in the literature of Montague grammar.

11. Pattern composition defines objects by induction rather than deduction or production in the sense of Post (1943). For conceptual consistency, I say that generative grammars are "inductively generative" grammars if and only if they generate languages inductively. Inductively generated languages are defined as follows:

Language L is an inductively generated infinite set of sentences if and only if:

i. There exists a finite set of primitive sentences such that $L^{\circ} = \{u_1, ..., u_n\}$.

ii. There exists a deterministic recursive procedure I (call this induction) which recursively maps L^i onto L^{i+1} $(0 \leq i)$ so that $L^i \neq L^{i+1} = \{L^i, C^{n,i}\}$, where $C^{n,i}$ is n-ary composition out of n sentences in L^i, assuming $L^{\circ} \neq L^1 = \{L^{\circ}, C^{n,\circ}\}$, where $C^{n,\circ}$ is n-ary composition out of n sentences in L°.

iii. $L = L^k$ $(n < k)$.

Clearly, inductively generative grammars contrast deductively generative grammars in which production (or rewrite) rules are used to map a sentence to another; deriving all sentences from the so-called "initial symbol" *S*. In deductively generative grammars, no derivation in this sense takes place. Rather, they map L^0 up to L^k, namely map a language to another.

12. This is noted because the notion of notational variant is useful frequently without any empirical content. First, there is no explicit **definition** of the notion. Second, if, as Chomsky once remarked, generative semantics is a notational variant of standard theory, then it is unreasonable to state that differential calculus in Leibnizian notation (i.e., $f'(x) = dy/dx$) is a notational variant of Newtonian notation (i.e., $f'(x) = \dot{y}$), and likewise, (a) $\sin^2\theta + \cos^2\theta = r^2$ is a notational variant of (b) $x^2 + y^2 = r^2$. Such claims are of no use. Every mathematician will agree that Leibnizian notation is thousand times better than Newtonian (except for the purposes of saving paper). Moreover, notations (a) and (b) are based on different conceptions, and will lead one for different directions when thinking about the same problem: (a) is basically geometrical, while (b) is algebraic.

13. I do not claim, however, that children who learn a native language are exposed to sentenc-

es like this.

14. Whether *V* is lexical or functional is not clear.