Chapter 3

Conceptual Foundations of Pattern Matching Analysis

3.1 Introduction

In this chapter, I will give a synopsis of a method for syntactic analysis, to which I will refer as **pattern matching method for syntactic analysis**, or **pattern matching analysis** for short (PMA in abbreviation), which is proposed as a framework that provides a "realistic", "connectionism-savvy" description of natural language syntax.

The rest of this Section 1 lays out the general framework, by indicating motivations, background, basic assumptions, claims, and disclaimers. Section 2 introduces basics of PMA. Section 3 explores them and try to deal with the problem of syntactic structure, thereby arguing that syntactic structure, if any, is not adequately characterized as a tree structure. Section 4 makes clear some important implications in Section 3, and explores further consequences. Section 5 presents a pattern matching analysis of the "cross-serial dependency" in Dutch (Bresnan, *et al.* 1982), mentioning a word grammar analysis of it (Hudson 1984). Section 6 summarizes arguments of this paper.

3.1.1 Motivations

One of the central aims of pattern matching analysis is to **provide a realistic description of natural language syntax**. By description of language syntax, I mean **analysis of language syntax**, or simply **syntactic analysis**. PMA is motivated by the four specific theses as follows:

(1) I. **Description before explanation thesis.** Phenomena of natural language syntax, NL syntax for short, must be "described" before they are "explained". This motivation forbids us to appeal, in clearly circular fashion, to Universal Grammar to explain them.

II. **Proper description thesis.** If the goal is to offer an adequate description of NL syntax, it must be a proper one, and for this reason one may not "reduce" the phenomena of NL syntax to other aspects of language and/or cognition, no matter how clear their correlations are.

III. **Realistic description thesis.** A proper description of NL syntax may not be simply a "formal" one. A formal description only serves as a "caricature" of NL syntax.

III′. Furthermore, the realisticness of descriptions so provided must be as much "objective" as possible. This excludes rather "subjectively judged" realisticness of descriptions provided by most descriptions in terms of "image schemas".

IV. **Connectionism-compatibility thesis.** I take a realistic description of NL syntax to be compatible with as many connectionist results and theories as possible, and it must be so "substantially" rather than "terminologically", which some notable cognitive approaches seem to be. More explicitly, constructs used to describe phenomena of NL syntax should be "connectionist-compatible", and be "connectionist-savvy", if possible. I expect that the idea of subpatterns, which are a natural extension of "wickelphones" in the sense of Rumelhart and McClelland (1986), meets this requirement.

Some of those theses seem controversial, and let me make a few notes on controversial ones.

Syntactic analysis of NL may differ from semantic and pragmatic analyses of NL, on the one hand, and from phonological and phonetic analyses, on the other. More specifically, it is not expected that syntactic analysis quickly outlines any other kinds of analyses. It is not expected either that other kinds of analyses quickly outline syntactic analysis. At least superficially, they may be incompatible.[1] The strongest reason is that they are phenomena on different dimensions, on the one hand, and on different scales, on the other. Linguistic analysis should countenance this sort of **multiplicity and complexity in description** as many biologists do. No biologist would complain that a description of biological species at the level of "gene" does not reflect the description of it at other higher levels of expression, such as levels of growth, development, and behavior. Crucial here is that blind reductionism is unfruitful and even harmful in biology. I suspect that this should hold of linguistics.

Designed for description, the proposed framework, called pattern matching analysis, does not pretend to "explain" syntactic phenomena of language. It is often taken for granted, or even suggested, in the literature of linguistics that "explanatory adequacy" trades off with "descriptive adequacy". I find no grounds for such an idea. Good explanations, I have to note, do not precede good descriptions.

Keeping this in mind, I claim that, in pattern matching analysis, **providing adequate descriptions is primary**, and **providing adequate explanations is second-**

ary, even if it is possible. By this, though, I never suggest that the framework does not provide, or does not try to provide, any explanation. First, **what are deemed in linguistic literature to be explanations are mostly not explanations at all**. In fact, they are, in my view, mere descriptions.[2] Second, what is meant by the priority of descriptive adequacy is that if the framework will provide explanations, it will do so **indirectly**.

It should be stressed, then, that best descriptions should facilitate, or even bring with them, the best explanations. This **heuristic** aspect of descriptions should be emphasized. I hope that pattern matching analysis is a framework that offers such good descriptions that will provide us with insights into language syntax.

3.1.2 Background

A sort of developer's note may be helpful. Objects of inquiry at earlier stages of my research were somewhat different. I investigated by and large what I called **correspondence among surface formations** (Kuroda 1996, 1997), trying to define NL syntax as a system of conditions that license "local mismatches" such as the following:

(2)	*F*:	Jim	provided	food	for	them	•	•
				X	X		X	X
	G:	Jim	provided	•	•	them	with	food

Only partial mismatches are indicated by inserting X between, assuming that partial matches are default. • is a special symbol, called phantom, serving as a meta character that matches anything.

This is not intend as a description of a "transformation" from *F* to *G* or from *G* to *F*. Rather, "mutual support" is assumed between *F* and *G*. This means that both *F* and *G* are virtually base forms, with each a deep structure of the other. The model of syntax attacks on the "derivational" view of syntax.

This is partly due to the fact that earlier stages of my research were strongly influenced by Lakoff (1993), where he introduces **cognitive phonology**, thereby attempting, with success I believe, at partial elimination of derivations from phonology, thereby arguing against authors such as Bromberger and Halle (1989). Elimination is not total, however, because three levels (M-, W-, P-levels) are retained, and I was dissatisfied, especially in view of Karttunen (1993) who showed that Goldsmith-Lakoff three level system is superfluous. See also Goldsmith (1993) for allied research. In his attempt, Lakoff appeals to **(phonological) constructions** that are simultaneously satisfied well-formedness constraints between two levels (or within one level). Reinterpreting constructions, I thought that it was possible to characterize syntax in similar terms. Syntax may be, by and large, a system consisting of well-formedness statements among sentences in terms of correspondence. This is what I now call the idea of syntax (and semantics) external to sentence.

Later research has revealed, however, that such a picture was too simplified: the role of correspondence is indirect, and syntax (and semantics) internal to sentence is still necessary; or differently put "internal consistency" of sentences. The consistency is what I now call **pattern matching**. A rather drastic change in research direction took place when I discovered the method called **diagonalization** when I was working on **construction effects** (and resulted in Kuroda 1997), where I tried to offer an alternative to Goldberg's (1995) construction grammar account of the effects. My solution is **pattern blending**, whereby two surface form(ation)s are blended into one. I suggested that formations like H = *Joe kicked Bill the ball* are "blends" in roughly the following sense. Given F = *Joe kicked the ball* (= $X\ V\ Z$) and G = *Joe W Bill the ball* (= $X\ W\ Y\ Z$), H = *Joe kicked Bill the ball* (= $X'\ V'\ Y'$ Z' is a blend of F and G in that $X' = \zeta(Joe, Joe)$, $V' = \zeta(kick, W)$, $Y' = \zeta(\bullet, Bill)$, and $Z' = \zeta(the\ ball, the\ ball)$, where ζ denotes the superposition operator. Clearly, superposition is a sort of unification. For illustration, I present below a table indicating how F and G are blended into H.

(3)	*F.*	*Joe*	*kicked*	•	*the ball*
	G.	*Joe*	*W*	*Bill*	*the ball*
	⇓	⇓	⇓	⇓	⇓
	H.	*Joe*	*kicked*	*Bill*	*the ball*

Most effects of ditransitive construction in Goldberg's (1995) sense can be described as the properties of G-form which serves as the "host" that provides H with the syntax and semantics of *X give Z Y*.

What is crucial in this blending account is the assumption that the meaning of *give* approximates the one of W in G. How to justify this controversial point, however? For discussion, I assumed that **generalization over co-occurrences** of X, Z and Y in *X give Z Y* takes place to yield a schematic representation of a "local syntactic environment", i.e., ***X W Z Y***, where ***X***, ***Y***, ***Z*** have full "lexical" specifications, whereas W is an "after-image" of *give*'s occurrence which lacks phonology. Such an artifactual construct as W is natural when the associative nature of the brain is recalled.

Generalizing this line of thought, I soon realized that what are blended in such a way need not be "sentences" at all. They can be any "parts" of a sentence. Specifically, as many **subpatterns** as required can be blended into a sentence in such a way. This is how **pattern composition** was born.

3.1.3 Basic assumptions

Before starting, let me explain briefly what pattern matching analysis will and will not assume. Crucial assumptions are:

- What are usually called "surface forms" or "surface formations", or even "linguistic expressions" in general, are best characterized as "patterns" that are composed out of, and decomposed into, a set of subpatterns in the sense defined later.

- Very roughly, what are called "words" (and "constructions") are special cases of subpatterns in the sense defined later.
- Subpatterns are "schemas" which combine by themselves.

These considerations contrast the proposed view of words with a dominant view of them. According to the dominant view, surface form(ation)s are surface structures or s-structures that are "derived" from deep structures or d-structures, that the base rules of the grammar generate without making reference to the contents of the items, e.g., words, to be inserted into them.

Descriptions of syntax that pattern matching analysis provide do not crucially rely on the notions of deep structure and derivation from them. I do not deny some theoretical importance of such notions, but the notions play no significant role in the proposed framework. Technically, what I want to attack is the existence of the so-called "base component", which serves as the kernel of a generative grammar. My suggestion is that if there is something superfluous in Chomskian generative grammar, it is nothing but the base component. The base is entirely dispensable if words specify their syntax in addition to their meanings and pronunciations.

This view of words should not sound so strange. Everyone can agree that all word have pronunciations of their own and meanings of their own. If so, why is it impossible to assume that they have syntax of their own? The proposed framework tries to go along this line of thought.

So, if there is some structure that serves as a deep structure in classical sense, it is nothing but a set of "words" that comprise a surface formation.[3] But I should be more careful here. It is better to think that each such word specifies a structure that serves as a deep structure, and consequently a deep structure in classical sense should be equated with a "set of deep structures".[4] The key idea is that all words that comprise a (part of) surface form *F* are "partial descriptions" of the syntax, semantics and phonetics of *F*. This shows that the proposed framework embodies a "declarative" perspective on syntax (Bird 1995, Scobbie 1992, 1997).

3.1.4 Claims

Reflecting the assumptions above, pattern matching analysis refers to the nature of "knowledge of language" in the sense of Chomsky (1986*b*, 1988):

- Knowledge of language, equated with "grammar" in technical sense, ultimately reduces to **knowledge of words** (Hudson 1984, 1990).
- Words, in the sense of the proposed framework, are not mere words; they are themselves **schemas**, not only in the sense of cognitive linguistics (Lakoff

1987; Langacker 1987, 1991*a*, *b*), but also in the sense of neo-Piagetian school (Arbib 1989; Arbib, *et al.* 1987; Arbib and Hill 1988). To make this point clearer, I will refer to relevant units of grammar as **words as schemas**.

- Under these assumptions, pattern matching analysis is designed to provide a detailed **emergent theory** of syntactic structures.[5] In this framework, syntax emerges out of the interaction, or even communication, among word schemas (influenced by "connectionist" ideas, in particular those in Elman 1990, *et seq.*; Rumelhart and McClelland 1986).

Loosely, the proposed framework is more compatible with the trends in cognitive linguistics than those in Chomskian linguistics, with the note that not all research efforts in Chomskian linguistics are generative, and not all research efforts in generative linguistics are Chomskian.[6]

3.1.5 Disclaimers

It should be noted that the proposed framework does not subscribe to either view of the syntax-semantics interaction in the following:

A. Syntax defines structures only by which semantic and phonological structures are sanctioned.

B. Semantics (with (part of) pragmatics therein) defines structures only by which syntactic and phonological structures are sanctioned.

The view characterized in A is favored by many Chomskian and generative linguists. The view stated in B is favored by many linguists of functional and/or cognitive linguists.[7]

Pattern matching analysis disclaims both sorts of implications, partly because it is unqualified to claim one: the roles of semantics and phonology are "implicit" in the proposed framework.

I assume, quite conservatively, that grammar is a system that emerges out of the complex interaction among syntax, semantics, pragmatics, and phonology/phonetics. This view is conservative, and in a sense not so illuminating, but I find it is well motivated. I find no decisive evidence for or against either kind of view. Both are quite simplistic, thereby underestimating the real complexity of language. I never expect that reduction in the number of components and/or modules approaches the reality, and provide good results. In my view, the interaction among syntax, semantics, and phonetics/phonology is complex, perhaps more complex than believed.

3.2 Basics of Pattern Composition and Decomposition

Syntax is concerned with the problems of **composition** and **decomposition**. Obviously, most linguistic units are composed out of smaller units. For instance, sentences are composed out of phrasal units such as subject noun phrases, verb phrases, object noun phrases, preposition phrases, on the one hand, and out of words, on the other, provided that phrasal units in turn are composed out of words.

Both composition and decomposition are based on a crucial assumption: linguistic forms or formations are composed of **optimally meaningful units**, which can be very roughly equated with words. Thus, it is the greatest concern of a syntactic theory to explain how such units are composed out of smaller units. My argument begins by questioning this because it is not a trivial matter.

3.2.1 Patterns and their subpatterns

Let me begin by reviewing briefly the way (4), which I call a **surface formation**, is composed out of, and decomposed into, subpatterns.

(4) *Bill hates gates.*

In pattern matching analysis, the syntactic structure of (4) is roughly encoded by appealing to an encoding scheme, which I call either a **co-occurrence matrix** or a **composition/decomposition table** (C/D table for short) interchangeably.

(5)	0.	***Bill***	***hates***	***gates***
	1.	***Bill***	*V*	*(O)*
	2.	*S*	***hates***	*O*
	3.	*S*	*V*	***gates***

Descriptions in terms of encoding scheme like this are exactly what I want to replace descriptions of syntactic structures in terms of a tree. Leaving crucial details for later discussions, let me note a few most important points here.

Let me establish terminology. I will call the construct indicated by 0 (= ***Bill hates gates***) a **base pattern**. I call the constructs indicated by 1, 2, and 3 **subpatterns**. As noted above, subpatterns are partial descriptions of the base pattern.

A subpattern consists of two different kinds of components, **anchors** and **glues**. ***Bill***, ***hates***, ***gates***, which are boldfaced, are instances of the first kind, and *S*, *V*, *O*, which I call (**pattern**) **glues**, are instances of the second kind. *S* and *O* correspond only very loosely to subject and object. It is an open question whether it is possible to identify *S* and *O* under such names.

Glues are controversial constructs, because, in my interpretation, they do not have lexically specifiable contents either semantically or phonologically. Glues are crucial because they encode syntax necessary for surface formation in a "distributed" fashion. Without them, base generation of phrase-markers becomes inevitable.

More specifically, subpatterns 1, 2, and 3 are **declarative statements of co-occurrence restrictions** imposed on boldfaced units.

(6)
i. Subpattern 1 states that ***Bill*** precedes a unit which is categorized as *V*, and another unit which, following the *V*, is categorized as *O*, if any.
ii. Subpattern 2 states that ***hates*** follows, or postcedes, a unit which is categorized as *S* and at the same time precedes a unit whose category is *O*.
iii. Subpattern 3 states that ***gates*** postcedes a *V* which postcedes an *S*.

Obviously, *S V* ***gates*** specifies the position of *gates* rather **redundantly**. Note that it is possible to replace *S V* ***gates*** by *V* ***gates***, the latter of which encodes necessary and sufficient information for co-occurrence of *gates*, if it is coupled with ***Bill*** *V (O)* and *S* ***hates*** *O*. If *X* is *V*, then *S V X* automatically follows only if *S V* is stated elsewhere, as in (5)1. Likewise, if *X* is *S*, then (part of) *X* is *N*, with the unit with label *N* is the "head" of *S*.

Such simplification is possible, and usually is argued for by linguists. But it is incompatible with connectionist results and theorizing, on the one hand, and it is not factually necessary at all, on the other. **Specifications may be redundant**.

Indeed, pattern matching analysis is one of the frameworks in which this kind of prima facie simplification is judged to be "artifactual". Despite some linguistic theories supporting it for the **economy of description**, PMA will not attempt to eliminate such redundancy. Rather, PMA will try to make use of this kind of redundancy in specification. Redundant specifications are allowed unless they lead to contradictions with other specifications.[8] My point is two-fold: (1) economy of description counterbalances with **reality of description**, at least as far as the compatibility with connectionist results is concerned; (2) redundancy in specification is exactly the source of the **emergence** of syntactic structure.

3.2.2 Relation of conceptual and phonological information to relational information

Let us turn to another point of the encoding scheme in (5). The scheme is conceived to capture both composition and decomposition of (4). For composition, (5) claims that ***Bill hates gates***, in 0, is a "base" pattern that is composed out of subpatterns ***Bill*** *V (O)*, *S* ***hates*** *O*, and *S V* ***gates***, encoded in 1, 2, and 3. For decomposition, it claims that the base pattern is decomposed into the subpatterns.

Subpatterns comprise abstract constructs *S*, *V*, *O*, called (pattern) glues. Here and elsewhere, symbols *S*, *V*, and *O*, in nonboldface, mnemonically encode subject, verb, and object very loosely. As I noted earlier, glues are not merely grammatical categories, nor grammatical functions in usual sense. The role of glues that surround a word is schematic encoding of its context.

In this sense, glues cannot be meaning-free, and it is better to think that glues have syntax and semantics of their own, no matter how abstract they might be. Indeed, semantics of glues are hardly conceptual and rather relational. Contrasted to such glues are **anchors** of subpatterns like ***Bill***, ***hates***, ***gates***, which are always put in boldface. They specify conceptual and phonological contents of lexical items. Such contents could be referred to as **substantials**.

3.2.3 Abstractness of syntactic patterns

What is the relation of substantial terms to relational terms, then? A tentative answer is that there are such **schemas of syntax** as 1 = ***S*** *V (O)*, 2 = *S* ***V*** *O* and 3 = *S V* ***O***, as follows:

(7)	0.	*S*	***V***	***O***
	1.	***S***	*V*	*(O)*
	2.	*S*	***V***	*O*
	3.	*S*	*V*	***O***

Schemas here impose grammatical functions *S*, *V*, and *O*, on syntax-free words.

Without positing such schemas, one aspect of linguistic creativity could not be explained. It would be impossible for speakers to use new words that they have not experienced yet, and therefore have no knowledge of contexts in which they occur. Thus, acquisition of a language, or internalization of a grammar is, in our view, equated with master of the way to manipulate schemas of the kind specified by 1, 2, and 3 in (7), with varying degrees of schematicity, whose specification of variables is rich enough to categorize any new words.

3.2.4 Experientially basedness of abstract patterns

It should be emphasized that schemas like ***S*** *V (O)*, *S* ***V*** *O*, *S V* ***O*** above are **surface-true generalizations** that learners of a language can arrive at by exposing themselves to positive evidence alone. In this respect, acquisition of a language is **experientially based** par excellence. In emphasizing this point, PMA is in conformity with claims by cognitive linguists like Lakoff (1987), Yamanashi (1995, 1999*a*, *b*), on the one hand, and by connectionists like Elman, *et al*. (1995), and cognitive scientists like Arbib (1989), Arbib, *et al.* (1987), Arbib and Hill (1988), on the other.

I emphasize that pattern matching approach to the "logical problem of language acquisition" depends crucially on a specific conception of linguistic knowledge. I hold that there must be sever restrictions on representation of linguistic units. This in fact forms a conceptual link to connectionism. I will discuss this issue in Appendix B in greater detail. To mention a few points in advance, I claim, in conformity with Elman (1990, *et seq.*), that basic properties of language syntax are learnable as far as units of language are learnable, and learnable units are, connectionist results suggest, context-sensitive, redundantly specified units.

Thus, it would be reasonable to think that phonological and conceptual contents of words are something on which schemas like those in (7) are superimposed, where phonological and conceptual contents are encoded by rows prefixed by #:

(8)	0.	***Bill***	***hates***	***gates***	⇔	#0.	{*Bill*,	*hates*,	*gates*}
	1.	***S***	*V*	*(O)*	⇔	#1.	*Bill*		
	2.	*S*	***V***	*O*	⇔	#2.		*hates*	
	3.	*S*	*V*	***O***	⇔	#3.			*gates*

Bill, *hates*, and *gates* on the right specify the phonological and conceptual contents of ***Bill***, ***hates***, and ***gates***, on the left.

The relation of #1 and 1, for example, is orthogonal. In other words, the most fundamental aspect of syntax is reducible to neither phonology nor semantics (in particular, equated with conceptualization).

The crucial point may be characterized differently. One may say instead that **subpatterns** (roughly, words and constructions) **are schemas**; this statement should be true not only in the narrow sense used in cognitive linguistics literature, but also in the broader sense of Arbib, *et al.* (1987) and Arbib (1989).

Subpatterns ***Bill*** *V (O)*, *S* ***hates*** *O*, and *S V* ***gates*** are schemas learned through abstraction of itemic, "token-based" patterns such as:

(9) S. ***Bill*** *hate gates*, ***Bill*** *danced*, ***Bill*** *kicked a dog*, ...

V. *Bill* ***hates*** *gates*, *Ann* ***hates*** *pizza*, *Everyone* ***hates*** *Bill Gates*, ...

O. *Bill hates* ***gates***, *Ann dates at* ***gates***, *Carol saw great* ***gates*** *of Kiev*, ...

It is clear thus that in PMA, if syntactic structure "emerges", it is not in the sense of Langacker (1997) who, as recently mentioned, argues that constituent structure emerges when "conceptual groupings are symbolized by phonological groupings". In statements like this, his use of *emergence* is utterly superficial; it has nothing to do the modern sense of the term to cover emergence of "dissipative" structures in the sense of Nicholis and Prigogine (1989) and similar structures that "emerge" in complex systems.

I claim that syntactic structure is an emergent property; but my reason is that pattern composition and decomposition are natural properties of a process of self-organization that takes place in human brain. Or differently put, it is a **cooperative computation** in the sense of Arbib, *et al.* (1987) and Arbib (1989). Reinterpreting such works, I claim, somewhat metaphorically, that words and constructions are **agents** who interact with each other by imposing their selectional restrictions on each other.

Syntactic structure, under the conception of the sort describe above, can be characterized as a network of pairwise interrelations. To see this, it would be helpful to take for example the composition and decomposition of an abstract pattern 0 = 123 below:

(10)	0.	1	2	3
	1.	1	$r_{1,2}$	$r_{1,3}$
	2.	$r_{2,1}$	2	$r_{2,3}$
	3.	$r_{3,1}$	$r_{3,2}$	3

It is easy to see that base pattern 0 = 123 is represented by the digraph in Figure 3.1, and subpatterns 123, 123, and 123 by the digraph in Figure 3.2.

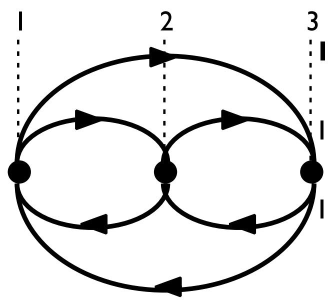

Figure 3.1

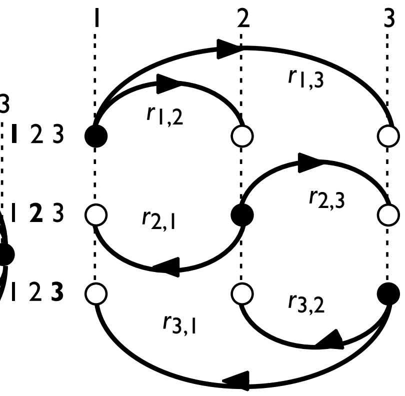

Figure 3.2

3.2.5 Details of pattern decomposition

Without specifying technical details, I simply state that the representation of *Bill hates gates* in (5), repeated here, is obtained by converting (by hand and mind, unfortunately) the itemic encoding in (11), which is obtained by **diagonalization** of the base pattern.

(5)	0.	***Bill***	***hates***	***gates***
	1.	***Bill***	*V*	*(O)*
	2.	*S*	***hates***	*O*
	3.	*S*	*V*	***gates***

(11)		1	2	3
	1.	***Bill***	*hates*	*gates*
	2.	*Bill*	***hates***	*gates*
	3.	*Bill*	*hates*	***gates***

It is vital to note that the i^{th} subpattern always corresponds to the i^{th} unit of a base pattern; and an arrangement of subpatterns 1, 2, ..., n always forms an $n \times n$ matrix.

A procedure, loosely called **categorization**, is clearly necessary, by which type-based encoding like the following is arrived at, starting from the token-based encoding in (11).

(12)

	1	2	3
1.	***Bill***	*V*	*(O)*
2.	*S*	***hates***	*O*
3.	*S*	*V*	***gates***

The three subpatterns 1, 2, and 3 above should be interpreted as follows:

(13)

	1	2	3
S =	***Bill***	*V*	*(O)*
V =	*S*	***hates***	*O*
O =	*S*	*V*	***gates***

This means that (4) = *Bill hates gates* is determined by a set $\mathscr{R}$, called the **role set** such that $\mathscr{R}$ = {*S*, *V*, *O*, ...} where *S* = ***Bill*** *V (O)*, *V* = *S* ***hates*** *O*, and *O* = *S V* ***gates***.

3.2.6 Syntax is a correlation of paradigmatics and syntagmatics

For more clarity, turn to the question, What do descriptions in terms of co-occurrence matrix really imply? It seems plausible to interpret that a co-occurrence matrix M denotes a correlation $M = {}^{t}R \times P$, where P and R are orthogonal arrays such that P = [..., 1, 2, 3, ..., n] and R = [..., S, V, O, ...]. P encodes a set of (relativized) positions in temporal dimension; and R a set of grammatical roles such as subject, verb, object. P correlates with time, and R with space.

Under this interpretation, it is reasonable to state that the proposed framework offers a new way of analyzing the syntagmatics/paradigmatics interaction. In other words, how syntagmatic and paradigmatic dimensions of language are interconnected, or how language organizes "time" and "space", by integrating them.

3.2.7 Pattern composition

So far I have described how a pattern is decomposed into a set of subpatterns. Turn now to the issue of how a pattern is composed out of subpatterns obtained by the decomposition sketched so far.

Pattern composition is obviously the inverse function of pattern decomposition. This must be the case; but it is nevertheless relevant to see under what conditions pattern composition is carried out.

Pattern composition is successful if and only if superposition of relevant subpatterns is successful. But under what condition does it take place? My answer is that superposition has to meet **type matching conditions**. To see this, return to (5), repeated here for convenience.

(5)	0.	***Bill***	***hates***	***gates***
	1.	***Bill***	*V*	*(O)*
	2.	*S*	***hates***	*O*
	3.	*S*	*V*	***gates***

Matching conditions are expressed very elegantly in this encoding scheme. Base pattern 0 is nothing but column-wise vertical **unification** of relevant units of subpatterns.

By way of example, ***Bill***, as the first unit of the base pattern is superposition of ***Bill*** in 1, *S* in 2, and *S* in 3. Likewise, ***hates*** and ***gates***, as the second and third units of the base, are superpositions of *V* in 1, ***hates*** in 2, and *V* in 3, and *(O)* in 1, *O* in 2, and ***gates*** in 3. They are unified column-wise.

It is inadequate to think that abstract elements like *S*, *V*, *O* called syntactic glues specify mere syntactic slots (into which lexical items are inserted). This can be interpreted as follows: (1) each glue is a mnemonic representation of **selectional restrictions** on a word. (2) what the glues surrounding a word specify is more generally an optimized description of syntax and semantics of a word's context. Noting that *Bill* ≠ ***Bill*** for example, ***Bill*** specifies, by notation, "substantial" component of word *Bill* with lexical meaning ***Bill*** *V (O)* (≠ *S V* ***Bill***). Slots *V* and *O* specify, on the other hand, "relational" components of the word *Bill*, which may have ideational (and even referential) contents of their own.

To summarize, symbols like *S*, *V*, and *O* are not meaning-free slots; rather, they are meaning-sensitive glues without which subpatterns are unable to combine to form a pattern without making reference to syntactic "templates" or "skeletons" like $[_S$ NP $[_{VP}$ V NP]] (adequacy of assigned categories is utterly irrelevant here).

From this follows the notion that the role of **syntactic categories** in syntactic description can be reduced drastically, if not completely. This is due to redundant specification allowed in PMA. Note that if unit *X* is a subject, it entails that *X* is a noun (phrase); but not vice versa. In other words, so-called syntactic categories like N(P), V(P), are simply too general to provide adequate description of language syntax.

3.2.8 The role of overlaps in pattern composition

Turn now to crucial aspects of overlapping among subpatterns.

Many linguists take it for granted to appeal to "templates" like $[_S$ NP $[_{VP}$ V NP]] to describe syntax of language. Remember, though, that it is not clear whether such templates are really necessary. It is in fact circular to think that they are provided by Universal Grammar.

In the proposed view, a surface formation, or a pattern, is decomposed into a set of subpatterns so that every one of them contains at least one glue. This is because, if this is not the case, it becomes impossible to dispense with externally defined templates like $[_S \text{ NP } [_{VP} \text{ V NP}]]$ to combine a subpattern with others.

The key to the emergence of surface formation is the **overlapping among subpatterns**, equated with words, on the one hand, and **redundant specifications in subpatterns**, equated with overlaps, on the other. Without them, pattern composition would never work adequately and efficiently.

What motivates this kind of redundancy, however? To make a long story short, my decision is to make PMA **connectionist-compatible**, which I find is a best way to make the proposed framework "cognitively realistic". I claim that PMA is connectionist-compatible, or at least **connectionism-savvy**, in that subpatterns is an extension of the idea of **wickelphones** proposed and used by Rumelhart and McClelland (1986) to encode a portion of the phonology of English. I added to wickelphones **schematicity**, realized by *S*, *V*, *O*. I will discuss this issue in greater detail in Appendix B.

To make relevant points clear, consider a simple example. Given a formation $F = u_1\ u_2\ u_3\ u_4\ u_5$, where u_i is the i^{th} unit of *F*. $[_1$ *Bob* $][_2$ *sliced* $][_3$ *bagels* $][_4$ *with* $][_5$ *a knife* $]$ is an example, though u_i need not be a word. Suppose then that there is a pattern *P* that matches *F* such that $P = C_1\ C_2\ C_3\ C_4\ C_5$ (e.g., NP V NP P NP) of preterminal symbols. C_i is the i^{th} unit of the sequence of preterminal nodes of *P*. Note that it is patterns, rather than trees, that match surface formations like *Bob sliced bagels with a knife*. This indicates that phrase-markers is virtually an "intermediate format" needed only to generate strings of preterminals $C_1 \cdots C_n$ (e.g., NP V NP P NP) as "templates", or rather "patterns" in the sense of PMA.

In (14), I show a case where five units, $u_{1,1}, u_{2,2}, \dots , u_{5,5}$, are "inserted" into, or attached to, the "terminal" nodes of $P = C_1\ C_2\ C_3\ C_4\ C_5$ (= 0). All units are context-free.

(14)

$$\begin{array}{llllll}
0. & C_1 & C_2 & C_3 & C_4 & C_5 \\
 & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\
1. & u_{1,1} & & & & \\
2. & & u_{2,2} & & & \\
3. & & & u_{3,3} & & \\
4. & & & & u_{4,4} & \\
5. & & & & & u_{5,5}
\end{array}$$

Arrows $\Uparrow$ here are interpreted as the operator of insertion or association of terminal lexical items to *P*.

This first case illustrated in (14) would best characterize the relation of lexical items to the phrase-marker as assumed by most generative linguists. They assume that the base component of a generative grammar generates strings of preterminals like $P = C_1\ C_2\ C_3\ C_4\ C_5$ (e.g., NP V NP P NP), on the one hand, and match them to strings of lexical items. For illustration, (15) gives the matching between NP V NP

P NP and [*Bob*][*sliced*][*bagels*][*with*][*a knife*].

(15)						
	0.	NP	V	NP	P	NP
		$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$
	1.	*Bob*				
	2.		*sliced*			
	3.			*bagels*		
	4.				*with*	
	5.					*a knife*

This conception is far from inevitable, however. Generation of base patterns is dispensable if subpatterns overlap with each other. To see this, let me examine a few cases. First, consider a case where four subpatterns, $u_{1,1}$ $u_{1,2}$, $u_{2,2}$ $u_{2,3}$, … , $u_{4,4}$ $u_{5,5}$, of length 2 are defined, each with overlap of length 1. The subpatterns are composed into a "complete" pattern at length of 5, in the following way.

(16)						
	0.	u_1	u_2	u_3	u_4	u_5
		$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$
	1.	$u_{1,1}$	$u_{1,2}$			
	2.		$u_{2,2}$	$u_{2,3}$		
	3.			$u_{3,3}$	$u_{3,4}$	
	4.				$u_{4,4}$	$u_{4,5}$

Here, 0 is obtained by column-wise unification of $u_{i.k}$ and $u_{j.k}$ into u_k by eliminating prefix indices *i* and *j*.

Remarks. $\Uparrow$ in this case denotes the operator of unification, while $\Uparrow$ in (14) is interpreted as the operator of insertion or association to $P = C_1\ C_2\ C_3\ C_4\ C_5$.

In this case and other similar cases discussed shortly below, base patterns need not be independently defined (by base rules, for example), as long as subpatterns 1, 2, …, 5 are already defined. This implies that syntactic structure can be "emergent" in that subpatterns specify informations of resulting structure only partially. In (16), base pattern 0 is "generated" by unifying overlapping subpatterns. This strongly suggests that it would be possible to dispense with the base component of a generative grammar altogether if a powerful learning mechanism is provided like recurrent networks discussed in Elman (1990, *et seq.*) and related works.

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.

For comparison, consider next a case where three subpatterns at length of 3, $u_{1,1}$ $u_{1,2}$ $u_{1,3}$, … , $u_{3,3}$ $u_{3,4}$ $u_{3,5}$, are given.

(17)
$$\begin{array}{llllll} 0. & u_1 & u_2 & u_3 & u_4 & u_5 \\ & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\ 1. & u_{1,1} & u_{1,2} & u_{1,3} & & \\ 2. & & u_{2,2} & u_{2,3} & u_{2,4} & \\ 3. & & & u_{3,3} & u_{3,4} & u_{3,5} \end{array}$$

In this case, too, base pattern "emerges" as subpatterns are composed. The same is true of the following case, where there are two overlapping patterns of length 4 given.

(18)
$$\begin{array}{llllll} 0. & u_1 & u_2 & u_3 & u_4 & u_5 \\ & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\ 1. & u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & \\ 2. & & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} \end{array}$$

The situation becomes different when there is only one subpattern, $u_{1,1}\ u_{1,2}\ u_{1,3}\ u_{1,4}\ u_{1,5}$ given. It is "vacuously" composed into the base pattern at length of 5, as the following illustrates:

(19)
$$\begin{array}{llllll} 0. & u_1 & u_2 & u_3 & u_4 & u_5 \\ & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\ 1. & u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} \end{array}$$

It can bee seen that the length (L) of each subpattern and the number (N) of subpatterns required to constitute the whole satisfies: $N + L > \max(N)$. Confirm this by seeing $(N{:}L) = (1{:}5)$ in (14), (2:4) in (16), (3:3) in (17), (4:2) in (18), and (5:1) in (19).

Interestingly, (14) and (19) are two special cases, but in two opposite ways. In (14), the subpatterns (= lexical items) have no overlaps. In (19), there is only one (vacuous) subpattern that is the same size as the whole. In both cases, the notion of a subpattern is nonsensical. In (14), subpatterns are all context-free, or rather "decontextuated" units. Such subpatterns, with no overlap, cannot combine with each other alone, and for this reason, a template $P = C_1\ C_2\ C_3\ C_4\ C_5$ is necessary. In (19), by contrast, the part and the whole are the same, and no composition is necessary.

Composition by superposition dispenses with any external machinery of composition. This is because the resulting structure is an emergent structure. So, one can dispense with (terminals of) phrase markers, e.g., $P = C_1\ C_2\ C_3\ C_4\ C_5$ in (14), generated at the base. Likewise, it is free from "image markers", posited in some cognitive approaches like Langacker's cognitive grammar (1997), by which structures at the phonological pole are determined.

To conclude, I give a pattern matching analysis of *Bob sliced bagels with a knife*. Elaborating necessary details, (20) gives a pattern matching analysis of the form.

(20)	0.	***Bob***	***sliced***	***bagels***	***with***	***a knife***
		⇑	⇑	⇑	⇑	⇑
	1.	***Bob***	*V*	*(O)*		
	2.	*S*	***sliced***	*O*		
	3.	S	*V*	***bagels***		
	4.	*S*	*V*		***with***	*O*
	5.	*S*	*V*		*P*	***a knife***

This analysis assumes three subpatterns (1, 2, 3) with overlaps of length 2 and two subpatterns (4, 5) with overlaps of length 3. Glues like *S*, *V*, *(O)*, *P* are used to encode overlaps among subpatterns, with required schematicity added.

3.3 What Structure Is Syntactic Structure, if It Isn't A Tree?

Before launching into detailed analyses, I will make a few general remarks on the possible form of syntactic structures.

The technical metaphor, SYNTACTIC STRUCTURE IS A TREE (STRUCTURE), has in fact played crucial roles in the development of syntactic theories since the Chomskian revolution that followed the publication of Chomsky's *Syntactic Structures* (1957). The metaphor is so overwhelming that everyone thinks of any structure of language in terms of a tree. But I do not accept this metaphor, because one of the motivations for PMA is exactly that tree-based description of syntactic structures is quite problematic. In fact, what was shown to date in linguistic literature, generative or not, is that tree structure is necessary for the description of syntactic phenomena. But it has never been shown that it is sufficient. Indeed, no justification has never been provided that syntactic structure is not more complex than tree structure. PMA explores the possibility that syntactic structure, if any, is more complex than tree structure.

3.3.1 Classical account of syntactic structure

According to the metaphor most frequently used since the birth of generative linguistics, syntactic structure is a **tree**. Here, a tree designates a directed, rooted, crossing-free, cycle-free graph (structure). For expository purposes, I give a "classical" tree representation to account for the syntactic structure of (21), as generated by production rules in (22) through derivational steps: S ⇒ NP VP ⇒ N VP ⇒ N V NP ⇒ N V N ⇒ *Ann* V N ⇒ *Ann bothers* N ⇒ *Ann bothers us*.

(21) *Ann bothers us.*

(22) i. S → NP VP, VP → V NP, NP → (D) N

ii. N → *Ann*, N(P) → *us*, V → *bothers*

The derivational steps can be diagrammed as follows:

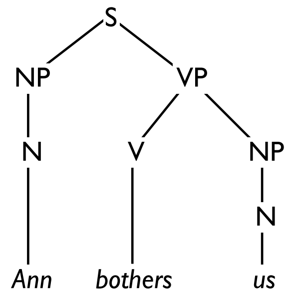

Figure 3.3

It is certain that such a derivation (from the initial symbol S) accounts for (21) because it is a string of terminal symbols. But, does this mean that the syntactic structure of (21) is adequately characterized by such kind of structure?

This is no definitive answer to this question, even if strong generativity is distinguished from weak generativity. For one thing, it is a theory that says that the syntactic structure of (21) is a tree such as in Figure 3.3, not a set of empirical facts. In fact, formations like *Ann bothers us* need not be generated by such a less powerful generative grammar which consists only of context-free rules in (22). Specifically, no empirical evidence has never been given that shows the following analysis is incorrect.

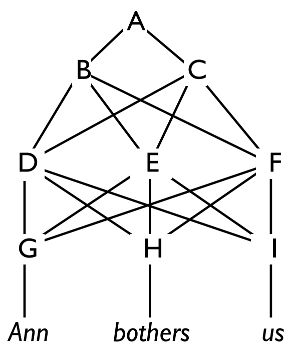

Figure 3.4

The structure diagrammed in Figure 3.4 is not a derivation tree, which linguists are accustomed to, but a derivation graph generated by a context-sensitive grammar.[9] Relevant production rules are as follows:

(23) i. $A \rightarrow BC, BC \rightarrow DEF$

ii. $D \rightarrow$ *Ann*, $E \rightarrow$ *bothers*, $F \rightarrow$ *us*

Note that $BC \rightarrow DEF$ is a context-sensitive rule because left-hand side of the rule contains more than one preterminal symbol.

In Figure 3.4, the difficulty in specifying $BC \rightarrow DEF$ in a graphical manner is handled by exhausting contributions of B and C to D, E, and F. Of course, this does not mean: $B \rightarrow DEF$ and $C \rightarrow DEF$. Rather, it describes a case of synchronized rewriting in which no rewrites take place independently.[10]

Apart from this hard problem, a natural interpretation of the rules in (23) would be: A = S, B (= D) = $\mathrm{N_SP}$ (subject noun phrase), C (= E) = V, F = $\mathrm{N_OP}$ (object noun phrase). This interpretation gives the following rule set.

(24) i. $\mathrm{S} \rightarrow \mathrm{N_SP\ V}$, $\mathrm{N_SP\ V} \rightarrow \mathrm{N_SP\ V\ N_OP}$[11]

ii. $\mathrm{N_SP} \rightarrow$ *Ann*, $\mathrm{V} \rightarrow$ *bothers*, $\mathrm{N_OP} \rightarrow$ *us*

A context-free grammar that does not include rules like $\mathrm{N_SP\ V} \rightarrow \mathrm{N_SP\ V\ N_OP}$ and therefore assigns derivation trees like the one given in Figure 3.3, rather than derivation graphs like the one given in Figure 3.4, is sure to be the weakest device to describe a set of strings like (21). But this does not mean that this has nothing to do with the question of whether the grammar of natural language, or more specifically the base component of it, is adequately characterized as such a context-free grammar. I will return to this issue in Section 3.5.5.

3.3.2 A new technical metaphor for syntactic structure

To illustrate crucial points, let us begin by asking how (21), for example, is composed out of the three words in the set (25).

(21) *Ann bothers us.*

(25) {*Ann*, *bothers*, *us*}

The relation of the units specified in (25) to (21) is exactly what is called syntax.[12]

A note: such a set as (25) can be taken to be part of **the lexicon**, in a variety of senses. I present the set as a "list" of lexical items only for expository purposes. In what follows, I will argue against the view of lexicon in which it is a list of context-free entries like *Ann*, *bothers*, *us*. More specifically, I will propose that what comprise the lexicon in a realistic sense are rather context-sensitive units like ***Ann*** *V (O)*, *S* ***bothers*** *O*, *S V* ***him***. Here, *S*, *O*, *V* are assumed to encode subject, object and verb, respectively, to which I will call subpatterns. Details of subpatterns will be discussed in Section 3.4.2.

Another, more important note is that syntax, in the sense defined above, has two different aspects. One aspect is **composition**, or **synthesis**, if the term is used loosely. Composition is conceived of here as the process and/or operation of "construction" of larger sized units, e.g., *Ann bothers us*, out of a given set of units of a given size. Another aspect is **decomposition**, or **analysis**. Decomposition is conceived of here as the process and/or operation of "deconstruction" of units of a given size into smaller units. We hold that the two aspects are distinct and may not be neutralized, contrary to the common belief. I do not detail my arguments here.

{*Ann*, *bothers*, *us*, ...} is a "vocabulary", e.g., a list of putatively "smallest" units. Mathematically, however, there are many other intermediate levels of analysis or synthesis between the largest and the smallest units. To see this, it is helpful to appeal to the notion of lattice. The diagram in Figure 3.5 illustrates the lattice of precedence relations compatible with *Ann* $\prec$ *bothers* $\prec$ *us*, with $\prec$ denoting the precedence operator.

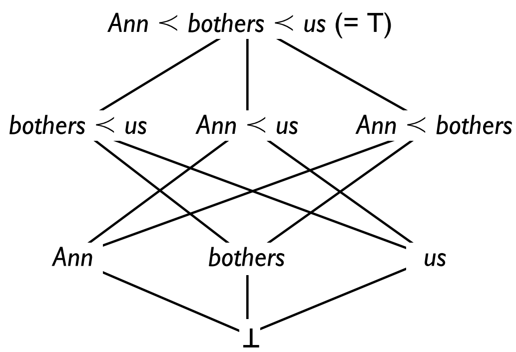

Figure 3.5

It is noticeable that the tree illustrated in Figure 3.6 is implicit in this lattice, if (i) *Ann bothers us*: S, (ii) *bothers us*: VP, (iii) *Ann*: NP, (iv) *bothers*: V, and (v) *us*: NP.

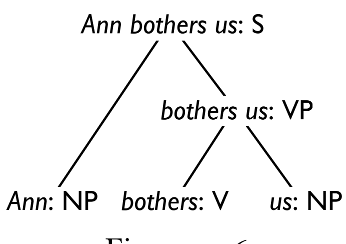

Figure 3.6

This fact strongly suggests that syntactic structure is more complex than a tree and less complex than a power set lattice. Unfortunately, I am not competent to formally prove this conjecture here.

Returning to the original question of how *Ann bothers us* = (21) is composed out of, and decomposed into, words *Ann*, *bothers*, *us*, in (25). Despite its superficial simplicity, even this question is deep enough to reject most answers unless the system by which they are arranged is properly specified.

3.3.3 Pattern matching perspective on composition

Without good empirical evidence, though, many linguists hold that "phrase markers" such as $[_{Y^2} [_{X^2} \ldots][_{Y^1} [_{Y^0} \ldots][_{Z^2} \ldots]]]$ are necessary to account for surface formations. Terminals of a phrase marker serve as "slots" into which lexical items, e.g., *Ann*, *bothers*, and *us*, are inserted (or associated). In this scenario, the **syntactic structure** of *Ann bothers us* is specified as $[_{Y^2} [_{X^2}\ \textit{Ann}\ ][_{Y^1} [_{Y^1} [_{Y^0}\ \textit{bothers}\ ]][_{Z^2}\ \textit{us}\ ]]]$, often depicted in the following way:

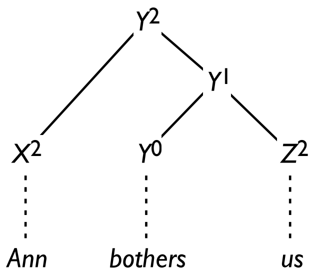

Figure 3.7

Usually, X^2 and Z^2 = NP (or DP), and Y^2 = S, Y^1 = VP, Y^0 = V (or Y^0 = IP, Y^0 = I^0). It would be helpful to compare this with the diagram in Figure 3.3.

It is inevitable in this conception of grammar to have a system of production rules (or principles) to generate structures such as $[_{Y^2} [_{X^2} \ldots][_{Y^1} [_{Y} \ldots][_{Z^2} \ldots]]]$, because no information necessary for this is provided in the list {{*Ann*, *bother*, *us*}, ...}, which is called the lexicon.

But the question still remains: Is such a system required independently of internal structure of lexical items? Or equivalently, Is it really adequate to think of the lexicon as a list of items which have no structural information?

The view of the lexicon that PMA advocates is quite different. Information that phrase-markers solely supply in a generative grammar is immanent in lexical information that words bear. Bits of such information are embodied in each word in a "distributed" fashion. This makes the base component redundant and useless.

Under the preliminaries above, I am now ready to specify another technical metaphor to capture syntactic structure. A surface formation is conceived of as a **synchronized pattern on a multi-tracked tape**, as music is.

For illustration, consider *Ann bothers us* in (21). The formation is metaphorically seen as a "play" of three subpatterns 1, 2, and 3 on three tracks, to be synchronized, or even "orchestrated".[13]

(26)	0.	***Ann***	***bothers***	***him***	Pattern played
	1.	***Ann***	V_1	(O_1)	Pattern played on Track 1
	2.	S_2	***bothers***	O_2	Pattern played on Track 2
	3.	S_3	V_3	***him***	Pattern played on Track 3

For purposes of exposition, I will refer to tables like this as **composition/decomposition tables** (or C/D tables for short), because they can be made neutral to the composition/decomposition distinction.

Note that synchronization plays a crucial role in pattern composition (and decomposition). To see this, let me rely on a more abstract version of (26), where $\boldsymbol{u}_i$ encodes the i^{th} unit of pattern 0, and $r_{i.j}$ encodes a role that its "context" determines. More explicitly, it is categorization relative to $\boldsymbol{r}_{i.i}$:

(27)	0.	$\boldsymbol{u}_1$	$\boldsymbol{u}_2$	$\boldsymbol{u}_3$	Resulting composed pattern
	1.	$\boldsymbol{r}_{1.1}$	$r_{1.2}$	$r_{1.3}$	Pattern played on Track 1
	2.	$r_{2.1}$	$\boldsymbol{r}_{2.2}$	$r_{2.3}$	Pattern played on Track 2
	3.	$r_{3.1}$	$r_{3.2}$	$\boldsymbol{r}_{3.3}$	Pattern played on Track 3

It is easy to see that (26) is an instantiation of (27) in that:

(28) i. $\boldsymbol{r}_{1.1}$ = ***Ann***, $\boldsymbol{r}_{2.2}$ = ***bothers***, and $\boldsymbol{r}_{3.3}$ = ***him***

ii. $r_{1.2} = V_1$, $r_{1.3} = (O_1)$, $r_{2.1} = S_2$, $r_{2.3} = O_2$, $r_{3.1} = S_3$, and $r_{3.2} = V_3$

Subpatterns 1, 2, and 3 are **synchronized** so that the index j of $\boldsymbol{r}_{i.j}$ encodes **timing of synchronization**, and accordingly all the units of subpatterns, with the index j, occur at the same time.

From the PMA perspective, words like *Ann*, *bothers*, and *us* comprising *Ann bothers us* are not mere lexical items. Rather, they are units structured so that phrase markers like $[_{Y^2}\ [_{X^2} \ldots\ ][_{Y^1}\ [_{Y^0} \ldots\ ][_{Z^2} \ldots\ ]]]$ are made unnecessary. More specifically, they are subpatterns to be synchronized with each other, specified as follows:

(29) i. ***Ann*** V_1 (O_1) for word *Ann* as a subject,

ii. S_2 ***bothers*** O_2 for word *bothers* as a transitive verb, and

iii. S_3 V_3 ***him*** for word *us* as an object.

3.3.4 Pattern composition by superposition

Turn now to the next question, How does composition operate? Let me explain this by looking at another example.

(30) *John ran rapidly.*

To compose this sentence out of words, *John*, *ran*, *rapidly*, I assume **unification** of three subpatterns ***John*** *V (O)*, *S* ***ran***, *S V* ***rapidly***. We also assume that unification needed to be synchronized. Thus, the base pattern is obtained by **superposing** subpatterns 1, 2 and 3 in the way illustrated below, where superposition is understood as a **column-wise unification** (indicated by ⇓).

(31)	1.	***John***	V_1	(O_1)	•
	2.	S_2	***ran***	•	•
	3.	S_3	V_3	•	***rapidly***
		⇓	⇓	⇓	⇓
	0.	***John***	***ran***	•	***rapidly***

In this table, "•", resulting from no match, serves as a neutral element, called "phantom", which is conceptually distinct from Ø for a zero form.[14] As a result of unification, the most faithful specification for the base pattern would be:

(32) $(John \times S_2 \times S_3)(ran \times V_1 \times V_3)((O_1) \times \bullet \times \bullet)(rapidly \times \bullet \times \bullet)$

Here, "×" indicates unification operator.

Thus, it is clear that *John* bears roles S_2 and S_3, imposed by subpatterns 2 and 3. Likewise, *ran* bears roles V_1 and V_3, imposed by subpatterns 1 and 3, and *rapidly* bears no role imposed by other subpatterns.[15]

Characterizations so far were abstract, and make it hard to imagine what happens in pattern composition. For a better understanding, it would be helpful to diagram relevant points of pattern composition in (31) as in Figure 3.8. Circles correspond to domains of synchronization.

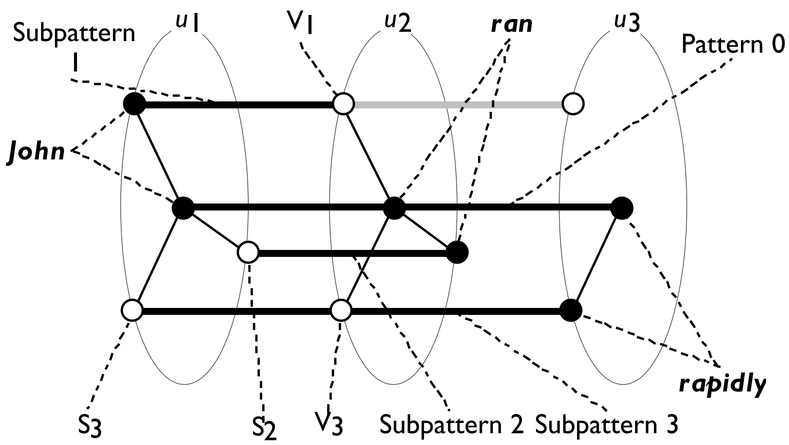

Figure 3.8

This figure shows the synchronization of the three subpatterns to result in *John ran* *rapidly*. In it, it is shown graphically how base pattern 0 = ***John ran rapidly*** is composed out of subpatterns 1 = ***John*** *V (O)*, 2 = *S* ***ran*** and 3 = *S V* ***rapidly***. Notably, the semantics (and phonology) of *John*, for example, results from unification of ***John*** in 1, S_2 in 2, and S_3 in 3.

3.3.5 Syntactic structure is more complex than a tree

What will happen, however, if it is tried to accommodate the tree metaphor to our multi-tracked tape metaphor? My suggestion is that tree-analysis is at best a partial characterization of a lattice structure that arises in pattern composition and decomposition, as shown Figure 3.9.

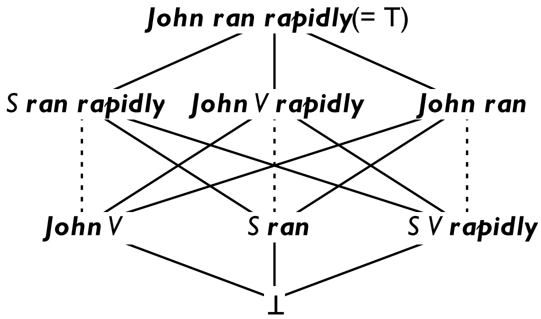

Figure 3.9. Dashed lines indicate complementarity pairs.[16]

There are some nodes that correspond to popular units. For example, *S* ***ran rapidly*** can be identified as a VP, though ***John*** *V* ***rapidly***, for example, has no obvious counterpart.

Comparing the diagram here with the diagram in Figure 3.7 above, it should be clear that the standard, tree-based conception of composition and decomposition embodies a few undesirable things. First of all, the composition of *John* and *ran* is made impossible unless *ran* is combined with *rapidly* before. There is, however, no empirical fact to necessitate this, though there a few kinds of fact that can be understood to suggest it. In addition, the composition of *John* and *rapidly* is impossible. As far as I can see, there is little empirical fact to motivate this, let alone necessitate it.

3.3.6 Pattern matching analysis of *V*-gapping

I would like to call attention to one point which I find is worth noting. There are certain unusual units like ***John*** *V* ***rapidly*** in the diagram in Figure 3.9, and, because of such exotic units, some kinds of deletion phenomena can be handled without appealing to the additional device of **deletion** or **movement**. Consider, for example, the following set of examples.

(33) a. *John ate quickly and* ***ran rapidly***.

b. *John ate quickly and* ***he ran rapidly***.

c. *Henry ran slowly, but* ***John ran rapidly***.

d. *Henry ran slowly, but* ***John rapidly***.

e. ?**Henry ran slowly, but* ***John ran***.

The problem is accounting for substrings like *ran rapidly* in (33)a, as contrasted with (33)b and c, on the one hand, and *John slowly* in (33)d, as contrasted with (33)c, on the other.

Most theories approach the phenomenon either by positing deletion of *John* or *he* in (33)b under (even sloppy) identity, or positing base-generation by which phonology-free slot (= *pro* or possibly *PRO*) is created in front of *ran rapidly*, though the latter will face problems in (sloppy) identity.

Admitting subpatterns like *S* ***ran rapidly*** and ***John*** *V* ***rapidly*** automatically serves virtually as an overgeneration-free base-generation account supplied with sloppy identity effect. To illustrate this, I give a pattern matching analysis to (33)b.

(34)	0.	***John***	***ate***	Ø[17]	***quickly,***	***and***	***he***	***ran***	***rapidly***
	1.	***John***	*V*	*(O)*					
	2.	*S*	***ate***	*(O)*					
	3.	*S*	*V*		***quickly***				
	4.	S_1	V_1		(AdV_1)	***and***	S_2	V_2	(AdV_2)
	5.						***he***	*V*	
	6.						*S*	***ran***	
	7.						*S*	*V*	***rapidly***

This is an *S V (AdV)*-conjunction,[18] which contrasts with a case of *V AdV*-conjunction:

(35)	0.	***John***	***ate***		***quickly,***	***and***		***ran***	***rapidly***
	1.	***John***	*V*	*(O)*					
	2.	*S*	***ate***	*(O)*					
	3.	*S*	*V*		***quickly***				
	4.		V_1		AdV_1	***and***		*V*	AdV_2
	5.	*S*						***ran***	
	6.	*S*						*V*	***rapidly***

Note that subpatterns 5 and 6 in the last analysis are virtually the same as subpatterns 6 and 7 in the one above.

3.4 Some Appeals of Pattern Matching Analysis

In this section, I will make explicit some consequences and interesting implications of the proposed framework.

3.4.1 Multiple parallel parsing as a communication among autonomous agents

Analogically, subpatterns are **autonomous agents**, running parallel in interaction with each other, to complete a composite pattern. This situation could be better characterized such that as many parsers as units run parallel. Thus, a pattern can be seen as a concurrence of finite state automata, with each equated with a subpattern.

3.4.2 Where do glues come from?

As noted above, what makes dispensable phrase markers and the base component altogether is the assumption that words themselves are **schemas** that have "syntax of their own". More exactly, words are themselves so structured that they comprise such components as *S* and *V*, to which I will refer as (**pattern**) **glues**.

Some remarks on glues will be helpful. First of all, what are glues? Where do they come from? My answer, inspired by connectionist results such as Elman (1990, *et seq.*), is that they are something that "emerge" as schematization of "contexts", or rather "co-occurrences", takes place. Schematization is based on the part/whole relation.

More explicitly, *S* ***ran***, the representation for *ran*, for example, is a generalization of its co-occurrences with other words. Part of them are:

(36) i. *John ran.*
i′. *he ran rapidly.*
ii. *Mary ran.*
iii. *cats ran.*

Generalization over such co-occurrences is the crucial mechanism. In more detail, I posit that learning English grammar is roughly a process of generalization in which a lot of disjunctions like (i) are successfully handled.

(37) i. {*John*, *he*, *Mary*, *cats*, ...} ***ran*** {Ø, *rapidly*}
ii. *S* ***ran*** *(AdV)*

Here, *S* describes the set {*John*, *he*, *Mary*, *cats*, ...}, and *(AdV)* the set {Ø, *rapidly*, ...}.

This view leads to important implications:

(38) I. Glues are "memory effects" of occurrence of words with other words.

II. Glues have abstract semantics and phonology of their own.

Thus, it is not adequate to regard them as meaning-free syntactic slots. So, glues like *S* and *O*, on the one hand, and *V*, on the other, only very roughly correspond to classes of NP and V, because glues are best characterized as **symbolizations of selectional properties**, perhaps in the same way that (phonological) segments are bundles of phonological features.

The basic idea is that a complex syntax emerges as lexical items interact with each other, imposing selectional restrictions of their own on each other. Glues encode necessary and sufficient information for such purposes.

Due to glues in them, words in our senses "know" what kinds of phrases their subjects and objects should be, at least metaphorically. One of its most crucial implications is that such subpatterns dispense with such phrase markers as [NP [V NP]], whether with S and VP labels or not, on the one hand, and dispense with even specifications like "*John* V(P)", "NP *ran*", as long as NP and V(P) are meaning-free, purely syntactic constructs. The strongest reason is that they are unnecessary because of their "overgenerality" with respect to ***John*** *V (O)*, *S* ***ran***.

Note that constructs such as ***John*** *V (O)*, *S* ***ran***, *S* and *V* need not be categorized more abstractly as NP, VP. Crucially, so-called co-occurrence restrictions play no role as far as selectional restrictions are observed; they are redundant in that they are always satisfied as far as selectional restrictions are satisfied. My interpretation is that co-occurrence restrictions in terms of *S*, *V*, *O*, are weakest expressions of selectional restrictions.

3.4.3 Mechanism of automatic checking

If the mechanism of content-sensitive superposition sketched above is to work properly, then the problem of **overgeneration** is drastically attenuated. In PMA, overgeneration should be prevented, at least in spirit, as composition, in the sense of derivation, of surface forms is not constrained by such and such constraints stipulated anyway. Rather, pattern composition is better viewed as a dynamic process of **self-organization** whereby pattern matching in itself serves as "automatic checking" device. Automatic checking takes place whenever pattern composition takes place as subpatterns are integrated. Incidentally, PMA is so-called basically due to this crucial role that this mechanism of pattern matching plays.

To clarify essentials, consider the following pair:

(39) a. ?**Colorless green ideas sleep furiously.*

b. *Revolutionary new ideas appear infrequently.*

While it is admittedly possible to give to (39)a an analysis like the following, it is impossible, or at least quite hard, to compose it coherently.

(40)	0.	***colorless***	***green***	***ideas***	***sleep***	***furiously***
	1.	***colorless***		N_1		
	2.	(AdN_2)	***green***	N_2		
	3.		(AdN_3)	***ideas***	V_3	
	4.			S_4	***sleep***	
	5.			S_5	V_5	***furiously***

The reason is simple and obvious. Note that N_1, for example, denotes a set of entities that "have colors", and clearly the set does not contain *idea(s)* in it. So, unification of N_1 and ***ideas*** blocks semantically. Similar blockages due to **type mismatch** should hold of pairs, (***colorless***, AdN_2), (N_1, ***ideas***), (N_2, ***ideas***), (V_3, ***sleep***), (V_5, ***sleep***), (S_4, ***ideas***), and (S_5, ***ideas***). So, it is more likely that (39)a is not composable (and not conceivable).

3.4.4 Comparison with *Optimality Theory*

In passing, let me make a few notes on the relation of PMA to *optimality theory* (Prince and Smolensky 1993). PMA is fundamentally different from optimality theory, despite the fact that it seems to be influenced and even inspired by connectionist theorizing. It should be made clear that my conception of pattern decomposition is inspired by Rumelhart and McClelland's (1986) idea of **wickelphonology**, which they developed owing to insights of Wickelgren (1969).

What makes optimality theory so powerful does not consist in its constraint-basedness; rather, its specific way of making use of constraints. Constraints in the theory are used as "output conditions" to "filter out" all undesirable forms. Basic insights into this sort of **surface/output filtering** go back to Ross (1967) and Perlmutter (1971).[19]

Optimality Theory needs a list of constraints to serve as output filters, and does not work without it. What determines generativity of grammar is ranking among such constraints, roughly analogous to modification of "connection weights" in localist neural networks. It is not clear, however, whether ranking and re-ranking among constraints in optimality theory is really an outcome of self-organization, since proponents of the theory seem to put no emphasis on learning process, over which connectionists had a debate with "classical" theorists like Fodor and Phylyshyn (1988) and Pinker and Prince (1988) (see Tesor 1995 and Tesor and Smolensky 1995 for an alternative point of view).

Let me conclude this digression by remarking that there is no denying that optimality theory is a promising linguistic theory, but it is somewhat unclear whether it can provides deep insights into human language, since, at least officially, it is a theory of competence, and at worst is another form of exploration into Universal Grammar, which might be a theory of irreality.

3.4.5 Further remarks on pattern composition

Since unification plays a crucial role in pattern matching analysis, it is clear that semantically, in *John ran rapidly*, *John* receives two (subtly different) semantic roles that subpatterns 2 and 3 impose on. *John* is understood as an agent based on the condition that it *S*-matches *S* ***ran***, where *S* denotes an agent. Likewise, *John* is understood as another agent based on the condition that it *S*-matches *S V* ***rapidly***, where *S* denotes an agent.

Essential points will be made clearer by considering another example.

(41) *John invited Ann to dinner.*

For this sentence, the following analysis is given.

(42)						
	0.	***John***	***invited***	***Ann***	***to***	***dinner***
	1.	***John***	*V*	*(O)*		
	2.	*S*	***invited***	*O*		
	3.	*S*	*V*	***Ann***		
	4.			*S*	***to***	*O*
	5.			*S*	*P*	***dinner***

Crucial here is the duality, or more adequately multiplicity, in thematic role that O_1 (e.g., *Ann*) manifests in construction $S\ V\ O_1\ (P)\ O_2$. *Ann* in this construction is understood as a "patient" (or "recipient") and at the same time as an "agent" (or "theme") of a certain motion. This kind of duality in NP's participant role has to do with so-called **subject-to-object raising**, broadly understood.

Raising is a phenomenon that is usually treated transformationally. It should be noted that raising analysis may work fine for cases like (43)a, whose putative underlying structure is (43)b, but it does not for cases like (41).

(43) a. *John found Ann (to be) boring.*

b. *John found Ann is boring.*

The reason is that (41) lacks a subordinate matrix that *Ann* serves as its subject unless preposition *to* forms such a matrix.

In PMA, the assumption is minimum: the two roles are imposed by *invite* and by *to*, respectively, by assigning *S* ***to*** O to preposition *to*. In handling cases like this, the postulation of a generalized class *R* = {*V*, *P*} effects adequately. It guarantees that *P* has a subject of their own. It is obvious that the effect can be accounted for if one simply assumes that S_4 ***to*** O_4 serves as an abstract kind of verb without appealing to syntactic movement or its conceptual analogue.

It should be also noted that *S V* ***Ann*** and ***Ann*** *V (O)* designates two different

types of *Ann* rather than two different **tokens** of *Ann*. While ***Ann*** *V (O)* designates *Ann* "as a subject", *S V* ***Ann*** specifies *Ann* "as an object".

Moreover, the dual role phenomenon in question is a side-effect of superposition of subpatterns, *S* ***invite*** *O* and *S* ***to*** *O*, the 2nd and 3rd subpatterns in (42), rather than a result of syntactic NP-movement (of *Ann*) from somewhere, possibly the *S*-position, if any (of *to dinner*) to the *O*-position (of *invite*).

3.5 On the Descriptive Power of Pattern Matching Analysis

Preliminary analyses presented so far seem to suggest that the weak generative power of pattern matching method exceeds the one of a context-free grammar, perhaps matching the one of a context-sensitive grammar, if not the one of an unrestricted grammar of the Chomsky hierarchy. I can only predict it because formal proof of the exact class of strings generated by the mechanism is beyond the scope of this present research. But, to support my prediction, I will present below a pattern matching analysis of so-called **cross-serial dependency construction** in Dutch (Bresnan, *et al.* 1982), which is known to be beyond the descriptive power of context-free grammars.

As I will show, the pattern matching analysis of the construction is straightforward. It is, in a sense, as much straightforward as Hudson's analysis of it in his framework of **word grammar** (1984, 1990) For expository purposes, I will compare my analysis with Hudson's, and suggest that the pattern matching analysis is "upper-compatible" with the word grammar analysis.

3.5.1 Basic properties of cross-serial dependency in Dutch

Cross-serial dependency construction in Dutch, illustrated in (44), is notorious since Bresnan, *et al.* (1982) called special attention for it.

(44)	... *dat*	*Jan*	*Piet*	*de kinderen*	*zag*	*helpen*	*zwemmen.*
	... that	Jan	Peter	the children	saw	help=inf	swim=inf
	'... that Jan saw Peter help the children to swim'						

This construction is important because it manifests a complex dependency in the subordinate clause introduced by *dat*. The substring there is said to show "cross-serial dependency" of the following sort:

(45) $a_1 \ldots a_n\ b_1 \ldots b_n$

We find a dependency between *a* and *b* such that the i^{th} occurrence of *a* corresponds to the i^{th} occurrence of *b*, or vice versa. Hence, the dependency is "cross-serial".

This sort of complexity in string cannot be handled by a context-free grammar, and this fact strongly suggests, Bresnan, *et al.* (1982) emphasize, that the class of natural languages is beyond the class of context-free grammar.

3.5.2 Pattern matching analysis of the cross-serial dependency

Pattern matching analysis of the cross-serial dependency construction is straightforward. Though quite tentatively, an analysis is given in (46).

(46)

0.	*S V*	***dat***	***Jan***	***Piet***	***de kinderen***	***zag***	***helpen***	***zwemmen***
1.	*S V*	***dat***	*S*			*V*		
2.		*(C)*	***Jan***$_i$			V_i		
3.		*(C)*		***Piet***$_j$			V_j	
4.		*(C)*			***de kinderen***$_k$			V_k
5.		*(C)*	S_i	O_j		***zag***$_i$	V_j'	
4.		*(C)*		S_j	O_k		***helpen***$_j$	V_k'
6.		*(C)*			S_k			***zwemmen***$_k$

To obtain this result,[20] extrinsic precedence statements as follows are needed:

(47) Dutch grammar stipulates that, in a subordinate clause introduced by *dat*:

i. all verbals (e.g., *zag*, *helpen*, *zwemmen*) postcede all nominals (or more weakly, *S* or *O*), keeping the relative serial order (e.g., ... zag_i ... $helpen_j$ ... $zwemmen_k$ ...), and;

ii. nominals arrange respecting "... *S* ... *O* ...".

In passing, I suggest that those statements are effects of *S V* which heads the construction. I suspect that this is a special case of **finite verb postposing effect** found in many Germanic languages. The effect may be roughly the "attraction" of *O* by *(C)*, and can be described using an anaphoric shifter, $C\ S\ \mathbf{O}_i\ V\ \delta(O_i)\ V$, probably licensed by *C*, thereby giving an alternative analysis such as follows:

(48)

0.	*S V*	***dat***	***Jan***	***Piet***	***de kinderen***	***zag***	***Ø***	***helpen***	***Ø***	***zwemmen***
1.	*S V*	***dat***	*S*			*V*				
2.		*(C)*	***Jan***			*V*				
3.		*(C)*	*S*	***Piet***$_j$		*V*	δ_i	*V*		
4.		*(C)*		*S*	***de kinderen***			*V*	δ_j	*V*
5.		*(C)*	*S*			***zag***	*O*	*V'*		
6.				*S*				***helpen***	*O*	*V'*
7.		*(C)*			*S*					***zwemmen***

It is an empirical problem whether (46) or (48) is correct, but (48) seems preferable in that it can dispense with extraneous indices. At any rate, I will not discuss this issue in this dissertation.

Whether (46) or (48) is correct, the pattern matching analysis of the cross-serial dependency construction is straightforward. No transformations are assumed to derive (44) as (a substring of) a surface form from an underlying form. A surface form is a surface formation that satisfies simultaneously as many constraints as words, since constraints are immanent in words.

3.5.3 Comparison with Hudson's *Word Grammar* Analysis

The straightforwardness of the pattern matching analysis is probably comparable with Hudson's analysis of it in his framework of **word grammar** (1984, 1990). For expository purposes, I will compare the two analyses.

Hudson (1984: 109) gives a word grammar analysis of the Dutch cross-serial construction in terms of **dependency structure**, as illustrated in the following, cited with some modifications.[21]

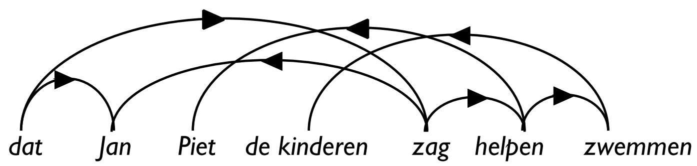

Figure 3.10

The assumed notational convention is that arrows go from licensers to licensees; or, in Hudson's terminology, from "heads" to "modifiers".

Some comments would be helpful. In Hudson's network, all grammatical relations are licensed by verbs or verbals, in this case, *dat*, *zag*, *helpen*, and *zwemmen*, rather than heads, and all licensing links start from them. This is because his theory is based on **valency**, a notion popular probably since Tesnière (1959).

The word grammar analysis of the cross-serious dependency structure, presented in Figure 3.10, is insightful, but I claim that PMA subsumes syntactic analyses provided by Word Grammar in terms of dependency structure. Below, I provide discussions to substantiate this controversial claim.

3.5.4 Diagramming co-occurrence matrices

My assertion that PMA subsumes WG analyses can be easily confirmed by checking that any dependency link in Hudson's diagram has a counterpart in the co-occurrence matrix in (46). In a sense, thus, a crucial property of pattern matching representation is that it makes explicit what dependency structure like in Figure 3.10 encodes implicitly. More specifically, I claim that the cross-serial dependency structure of (44) can be adequately diagrammed in the following fashion:

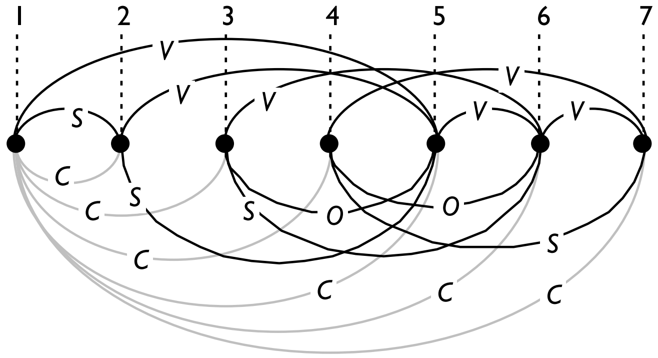

Figure 3.11

It is easy to see that this diagram encodes all dependency relations that Figure 3.10 encodes, with some additional relations. I will discuss them later.

Also, the diagram in Figure 3.11 represents only the "result" of pattern composition that is responsible for the dependency structure of (44), and it should be contrasted with the following diagram which represents the C/D table in (46) directly.

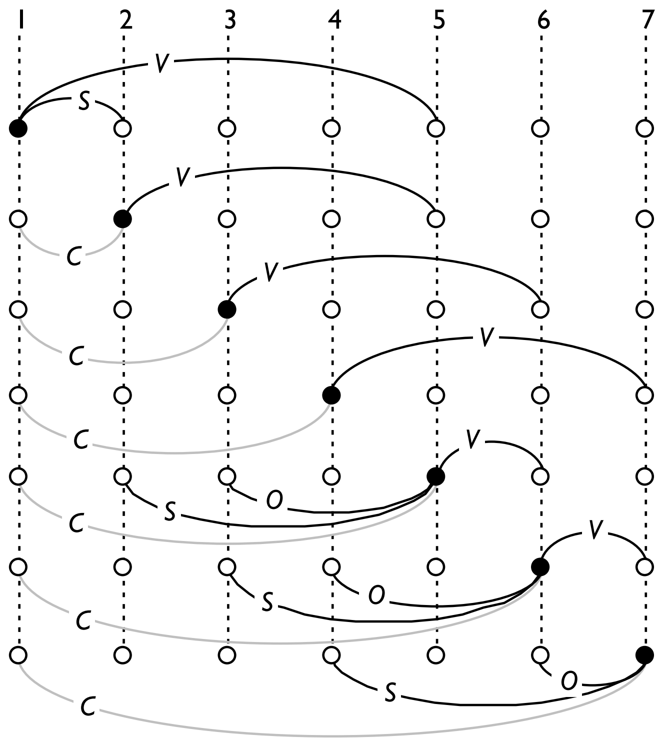

Figure 3.12

The two diagrams show more convincingly that the dependency structure in the sense of Hudson emerges as subpatterns combine by superposition.

The claimed upper-compatibility of the pattern matching analysis with Hudson's word grammar analysis would be guaranteed by the translation schemes of the following:

(49) i. Every link in Hudson's dependency structure is split into upper and lower sections. They can be interpreted as "demand" and "supply" in a communication network. Take for example, the link from *dat* to *zag*. The upper segment starts from a_1 = *dat* to reach e_1 (= node "S" in constituent analysis), and from there begins a lower segment, from e_1 to e_5 = *zag*.

ii. The direction of arrows is translated so that rightward arrows are in the upper triangle, and leftward arrows are in the lower triangle.

These rules are plain enough to ensure the upper-compatibility of pattern matching analysis to Hudson's word grammar analysis.

The last point proves an essential argument: those dependency diagrams like in Hudson's diagram in Figure 3.10 can never be more powerful than co-occurrence matrices like this. Moreover, co-occurrence matrices exhaust potential nodes for constituent analysis. This is obvious from the fact that co-occurrence matrix encodes all pairwise relations $r_{i,j}$ among units.

3.5.5 Is pattern matching a too powerful method?

The generative system that patter matching analysis assumes is powerful. This is good for "descriptive" purposes, since it will provide adequate descriptions for complex constructions, but not so for "explanatory" purposes.

For reasons that I can hardly accept, there are many linguists who do not prefer, or even disdain, grammars that have too much descriptive power, thereby trying to make grammars as little powerful as possible. They disfavor so-called "too powerful" grammars by holding as follows:

(50) A. Descriptions provided by such powerful grammars do not automatically lead to explanations.

I think this line of thought is unreasonable, and even absurd, because of a serious mistake. Such reasoning is groundless unless either thesis A or B in (51) is implicitly assumed, both of which sound absurd to me.

(51) B. Linguistics is a theory of grammars.

C. A linguistic theory is a theory of grammars.

Are these theses valid? Clearly, No. Either linguistics as a whole or a linguistic theory is a theory of grammars no more than physics is a theory of differential equations. Note that differential equations, with varying degrees of complexity, "describe" natural phenomena, and classes of them correspond to classes of natural phenomena in interesting ways. But this fact by no means indicate that (part of) physics is a theory of differential equations, which makes no sense. Such a theory is not part of physics but part of (pure) mathematics.

What natural scientists can hope is only that differential equations describe classes of natural phenomena very well, and not more than this. Note that this is just a hope because nobody can know why such and such differential equations describe such and such classes of natural phenomena. Such correspondences are immanent in all assumptions in physics, and can never be explained.

In a sense, linguists who hold the thesis stated in (50) try to "explain something that cannot be explained in any way". More specifically, it seems impossible to me to explain why only grammars of a limited complexity describes natural languages. It is a fact of nature. If some linguist says that is "explained" by Universal Grammar, I will say that such explanation is just a joke: it is not scientific at all. It is as much an awful joke as a theory of differential equations "explaining" Newton's equation $f = ma$.

I suspect that the source of such confusion lies in a serious mistake of descriptions for explanations among linguists. It is an empirical problem what complexity class subsumes the class of natural languages. It is reasonable that a proper subset of a natural language is a context-free language, whose strings are generated by a context-free grammar, even if the whole language is context-sensitive. But this fact does not mean that it is also reasonable to require that all rules of the base component are context-free, thereby making the transformational component alone a context-sensitive grammar, and keeping the base component within the class of context-free grammars. If the class is one of context-sensitive languages, which is suggested by a good deal of evidence, it makes no sense to discuss what component of grammar has context-sensitive generativity.

Note however that such requirements on the "architecture" of NL grammars are not empirically based at all. Again, there is no empirical evidence that all rules of the base component are context-free. Specification of the "weakest" generative system that gives adequate descriptions of all (and only) expressions of a natural language, which is one of the goal of generative linguistics, is still attainable even if the base component contains context-sensitive rules, and the transformational component is eliminated altogether. If base generation is context-sensitive, then I guess that (generative) grammars of NL is best characterized Lindenmeyer systems. In such systems, derivation can be seen as "growth". For more information about Lindenmeyer systems, see Ronzenberg and Salomaa (1980), Rozenberg and Salomaa, *eds.* (1992), and Vitányi (1980).

3.6 Concluding Remarks

In summary, claims about syntactic structure that pattern matching analysis has made have been presented as follows:

- Pattern decomposition of *F* is carried out by a special method of segmentation, called **diagonalization**, whereby an analysis of *F* is obtained nearly mechanically (but with some special assumptions) with respect to an arbitrary number of segments to be obtained.
- Decomposition of *F* results in a set, called **parse set**, consisting of *n* subpatterns if *F* segmented into *n* subpatterns.
- Any analysis obtained is not a "proper analysis", since subpatterns in a parse set thereby obtained **overlaps with at least one other subpattern**, if not all.
- Phrase-marking trees play no role. This is due to a special assumption assigned to diagonalization.
- Pattern composition is carried out by a method called **superposition** whereby subpatterns, roughly detailed descriptions of words, are superimposed over each other.
- We have a special descriptive device, called **co-occurrence matrix**, which captures correctly, and very insightfully, both how patterns are composed out of subpatterns and how they are decomposed into subpatterns.
- Pattern matching analysis is **connectionism-aware** in that what I will call subpatterns are a fairly natural extension of Rumelhart and McClelland's (1986) **wickelphones**.
- An assembly of subpatterns defines "syntactic structures", which are assumed to be **networks** of units rather than trees.

Pattern matching analysis is capable of serving a powerful tool for syntactic description of language.

Notes

1. In this regard, my position is similar to Sadock's framework called **Autolexical Syntax** (Sadock 1991), because he holds that morpho(phono)logy, or more adequately word-internal syntax, is "autonomous" to word-external (and sentence-internal) syntax, the latter of which is what is usually called syntax.

2. I have in mind here the case of "bounding theory" (Chomsky 1981), or subjacency condition in particular. The condition roughly says that move α may not move material at once over more than two bounding nodes, NP or $\bar{S}$ (or CP, especially in later reformulation of the effect in terms of "barriers" in Chomsky 1986*a*). Thus, the condition states, in an informal way, how syntactic movement is restricted. At least to those who are able to wonder if the naked king really wears clothes, however, it makes no sense to claim that the unacceptability of sentences like **Which woman did they believe a claim that everyone in the park laughed at?* is "explained" by this restriction on movement. My point is simple: **the subjacency condition merely describes, though with fairly accuracy, a specific restriction which is to be explained.** But this description promotes to an "explanation" as soon as Chomsky says it constitutes Universal Grammar. Here, circularity conquers all. Chomskians say that UG is concerned with competence rather than performance. All right, but if UG treats, by definition, matters concerned with "competence" only, then why is it concerned with subjacency condition? A real explanation must answer the question, Why may a moved material not jump over more than two bounding nodes? If one is able to wonder words of God Father of generative grammar, then he or she will soon realize that it is more likely that subjacency condition is **a constraint on performance**, and therefore it has nothing to do with UG, a theory of competence. In any case, this kind of explanations could not be regarded as "scientific" explanations. By this, however, that I do not suggest that subjacency condition is of little importance. What is of little importance, I claim, is rather UG, and any theory of competence. Note that it is better than anything that one can dispense with such a ghost of notion like competence. So, it is reasonable to think, quite realistically, that competence is unreal, or an abstraction that has nothing to do with linguistics of natural language. Universal Grammar has nothing to do with our reality of language any more than the Universal Turing machine has anything to with our reality of computer. So, our theory of language must be **UG-free**, since we conservatively assume that all we need is explanation of performance.

3. In some respect, there is a superficial similarity between recent minimalist program (Chomsky 1995) and the framework proposed here. Of course, the similarity is not more than a superficial one.

4. Multiplicity of the deep structure was suggested by Lakoff (1974) in his treatment of the phenomenon called syntactic amalgams. Kuroda (1999) proposes an alternative analysis for this phenomenon.

5. The use of "emergence" here is based on a crucial idea in Nicholis and Prigogine (1989) and related works in physics, in particular study of "chaos" (Gleick 1987). It may be somewhat misleading to associate with it the notion "emergent grammar" that Paul Hopper (1987) suggests in the context of grammaticalization.

6. I conceptually distinguish Chomskian linguistics from generative linguistics. Chomskian linguistics is a trend in linguistics that always appeals to an innate knolwedge system called Universal Grammar, which I find exists only nominally. By and large, I argue against Chomskian, UG-prone linguistics, despite my sympathy with generative linguistics.

7. This view also characterizes the generative semantics in late 60's and early 70's.

8. In this respect, the spirit of PMA is very much like the framework of **underspecification** in the sense of Archangeli (1984, 1988), with a charitable misunderstanding of the notion.

9. No straightforward interpretation of links between symbols at different derivational steps is possible for derivation graphs. It is not clear whether they indicate correspondence or not. Thus, two downward merging links from N_SP and V, at the second stage, into V, at the third stage, has no clear interpretation, even if the derivation is $S \Rightarrow N_SP\ V \Rightarrow N_SP\ V\ N_OP$.

10. This intepretation is based on my interpretation of parallel rewriting processes in **Lindenmayer systems**, or L systems for short.

An *L* system *G* is a triple $<\Sigma, P, A>$, where:

i. Σ is a set of symbols, without the distinction between terminal and preterminal symbols. Σ can be seen as a union of terminal and preterminal symbols such that $\Sigma = V_N \cup V_T$.

ii. P is a set of "productions" of the form $\alpha_1 \cdots \alpha_m \rightarrow \beta_1 \cdots \beta_n$ ($\alpha_i, \beta_i \in \Sigma$).

iii. A is a special symbol, called an "axiom", from which all derivations start. A may or may not be in Σ.

See Rozenberg and Salomaa (1980) for details of Lindenmayer systems. Let me remark on some relevant properties of L systems here.

Rewrites in a Lindenmayer system are said to be "parallel" in that all symbols of the input string have to be rewritten at the same time. For example, $G = \langle\{a, b\}, \{a \rightarrow aa, b \rightarrow bb\}, ab\rangle$ generates a language $\{ab, aabb, aaaabbbb, \ldots\}$, i.e., $a^{2^n}b^{2^n}$ ($n \geq 1$). This means that it is a natural interpretation to see derivational steps in L systems as "generations" in (cellular) development. Note that it has important effects if an L system has "vacuous" productions of the form $x \rightarrow x$. If $a \rightarrow a$, $b \rightarrow b$ are added to P in G above, the generated language no longer is $a^{2^n}b^{2^n}$; rather, it is a^+b^+.

Of linguistic interest is an extension of L systems with no specified axiom, which Vitányi (1981) calls L schemes. This allows the following reinterpretation of the $\overline{\mathrm{X}}$ scheme: (i) $X^0 \rightarrow Y^i X^1$, and (ii) $Y^i X^1 \rightarrow Y^i X^2 Z^j$, where Y^i and Z^j, serving as the specifier and complement of X, denote projections of X and Y up to the i^{th} and j^{th} generation. Assuming that S and O are of N^i and N^j, we can conceive of an L scheme that allows a "development" of V, such as $V^0 \Rightarrow S\ V^1 \Rightarrow S\ V^2\ O \Rightarrow S\ V^3\ O\ X$. The i^{th} row indicates the i^{th} generation of V. This allows to state that intransitive verbs are all verbs that cannot grow up to V^2.

11. This context-sensitive rule coulde be seen as a "merger" of the two rules: $\mathrm{N_SP} \rightarrow \mathrm{N_SP\ V}$ and $\mathrm{V} \rightarrow \mathrm{V\ N_OP}$.

12. More exactly, W-external and Σ-internal syntax, provided that W and Σ stand for word and sentence. Σ is used to distinguish it from S for subject.

13. I did not realize exactly what to replace trees by, until I read an illuminating passage in Bird (1995), who likened phonological organization to an "orchestral score". He employed the metaphor to characterize how "tiers" are structure in autosegmental conception of (morpho)phonology. For more information of autosegmental view, see Goldsmith (1979, 1990) and McCarthy (1981, 1984).

14. It is not clear, though, whether *(O)*×•×• really results in • . In this, • can be interpreted as a neutral element.

15. It is possible that tense-related features, implicit in *ran*, impose a role on *rapidly*.

16. Comlementarity is based on the part/whole relation.

17. Technically, the gap here is different from those that are created by movements, because its presence is not demanded by any unit in this case.

18. Despite supporting evidence, it is not clear whether *S V* constitues a single unit on a larger scale.

19. See Archangeli and Langendoen (1997) for a relevant discussion that contrasts optimality theory with Chomsky's minimalist program (1995).

20. This analysis avoids to appeal to shifters.

21. In his own diagram, what are linked are not words like *dat*, *Jan*, *Piet*, ... , but instances "1", "2", "3", ... (n^{th} word). This is necessitated by Hudson's distinction of word instances from words (as types). In word grammar, words are models, and word forms are instances of them.