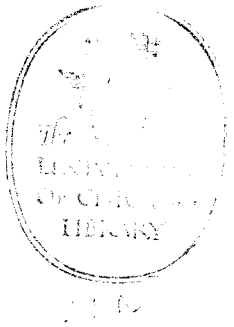
First published 1999

2 4 6 8 10 9 7 5 3 1

Blackwell Publishers Inc.
350 Main Street
Malden, Massachusetts 02148
USA

Blackwell Publishers Ltd
108 Cowley Road
Oxford OX4 1JF
UK



Library of Congress Cataloging-in-Publication Data
Crain, Stephen, 1947–
An introduction to linguistic theory and language acquisition / Stephen Crain and Diane Lillo-Martin.
p. cm. — (Blackwell textbooks in linguistics ; 15)
Includes bibliographical references and index.
ISBN 0–631–19535–1 (cloth : alk. paper). — ISBN 0–631–19536–X (pbk. : alk. paper)
1. Language acquisition. 2. Grammar, Comparative and general. 3. Psycholinguistics. I. Lillo-Martin, Diane C. (Diane Carolyn), 1959– II. Title. III. Series.
P118.C6747 1999
401'.93—dc21 98–34401
CIP

British Library Cataloguing in Publication Data
A CIP catalogue record for this book is available from the British Library

Typeset in 10 on 13pt Sabon
by Graphicraft Limited, Hong Kong
Printed in Great Britain by TJ International, Padstow, Cornwall

This book is printed on acid-free paper

Contents

16 A Transformation Generating Yes/No Questions

Introduction

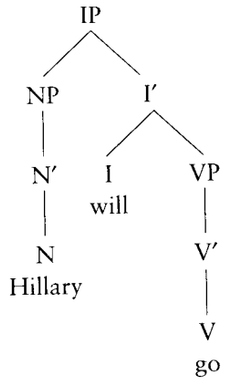
Phrase Structure rules alone can generate an infinite number of sentences, but they are not sufficient to generate all the sentence types found in English or any other natural language. In particular, we find that sometimes a relationship, or **dependency**, exists between elements in disparate parts of a sentence. For example, an element which is normally found following the subject (such as a modal) may be found preceding the subject in certain sentence types (such as yes/no questions). Dependencies are best captured by the use of **transformational rules**. In this chapter we begin our discussion of transformations. You will see that the concept of a transformation is relatively straightforward, although the implementation may become somewhat complex, especially when constraints on transformations are considered. It is worth going into this complexity as it helps to make the strongest case for innate knowledge.

1 Yes/No Questions

Consider the declarative statement in (1), which can be diagrammed using the tree in (2).

(1) Hillary will go.

(2)



This tree should be very familiar to you.

Now, consider the yes/no question in (3). (The question in (3) is called a "yes/no" question because the expected answer is either "yes" or "no.") Can this question be derived by our Phrase Structure rules?

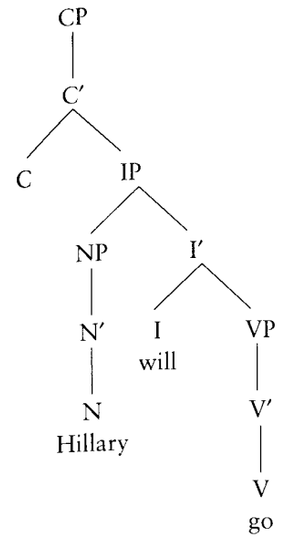
(3) Will Hillary go?

The X′-theory as we have implemented it in English cannot generate a sentence in which the subject comes after the element in I. It is possible to imagine ways in which to change our assumptions in order to generate this sentence using PS rules alone. For example, we could generate "will" higher in the structure, for example, in C, rather than in I. Or, we could generate the subject in the specifier of VP. However, there are reasons to take a different approach. Considering the similarities between the declarative in (1) and the question in (3), the grammar should capture the relationship between these sentences. Furthermore, we will see that the same kind of relationship exists between declaratives and other kinds of questions. In order to capture the generalizations these relationships exhibit, a different type of linguistic rule is used in addition to PS rules to generate questions such as that in (3). These are *transformational rules.*

A transformation begins with a well-formed **Deep Structure.** A Deep Structure is simply a phrase marker which is derived using the PS rules. It should always be compatible with these rules. The Deep Structure is the input to the transformation; the transformation then affects the Deep Structure in some way, producing a **Surface Structure** as its output. In this example (and in the example we turn to in chapter 18), the transformation moves an element from one position to another.

What is the Deep Structure for the sentence in (3)? It is essentially the structure of (1), but with one modification, anticipating the application of the transformation. The Deep Structure is given in (4).

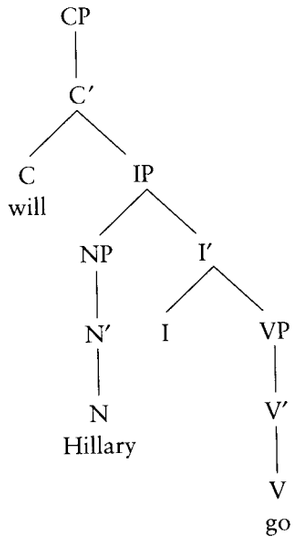
(4)



This phrase marker is very similar to the one in (2), and it is easy to see that it does follow the PS rules we have already presented (refer to chapter 8 if you have forgotten about CP). One special feature of this phrase marker is that the node C is present, but it has nothing beneath it. The node itself is not new: C was needed earlier in generating sentences such as "Dana is hoping that Kim will cook dinner." The word "that" which begins the subordinate clause, "that Kim will cook dinner," is dominated by the category C. Usually, we do not leave an empty node position in a tree (e.g., we routinely leave off the Spec of VP and PP). However, because we will use this tree as the Deep Structure for a transformation which will need the C position, we included it here. From now on, when an empty node is needed later in a derivation, it will be inserted at Deep Structure. In any case, it is clear that the phrase marker in (2) can be seen to follow from the usual set of PS rules within X′-theory.

How does a transformation transform the Deep Structure in (4) to the Surface Structure which goes with the yes/no question in (3)? The transformation, which we will call "Subject–Auxiliary Inversion" (or "SAI"), moves the contents of I to the position of C, resulting in the Surface Structure given in (5). (This transformation is also sometimes called "I-to-C movement," for obvious reasons.)

(5)



It is important to remember that the structure in (4) is the *Deep Structure* of the question in (3), because it follows the PS rules, and the structure in (5) gives the *Surface Structure*, which has the words in their appropriate places. Notice as well that (2) is the Deep Structure and the Surface Structure for the declarative in (1). That is, any sentence for which no transformation applies has the same Deep and Surface Structures.

Let's try another example. How is the question in (7) related to the declarative in (6)?

(6) Al is speaking.

(7) Is Al speaking?

The answer is diagrammed in figure 16.1.

Thus, we have discovered how to derive a "yes/no question." First, a Deep Structure is generated using the usual PS rules. Next, the transformation of SAI applies, moving the contents of I to C, resulting in the Surface Structure.

2 Affix-Hopping and Do-Support

Let's now consider what happens to create a yes/no question from a declarative like the one in (8).

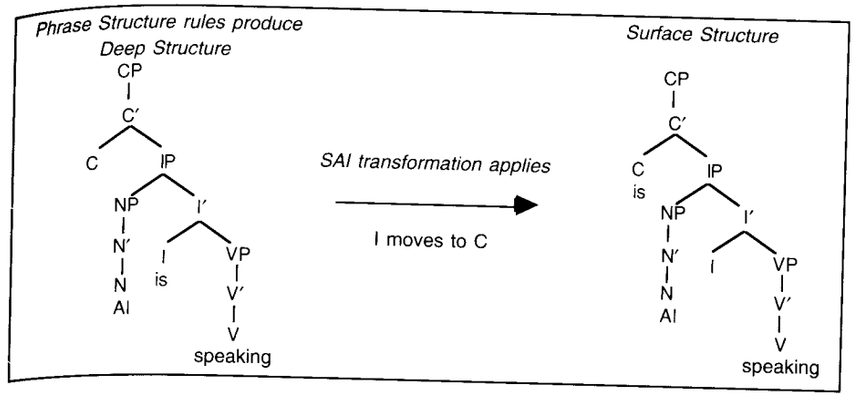


Figure 16.1 The derivation of a question

(8) Bill left.

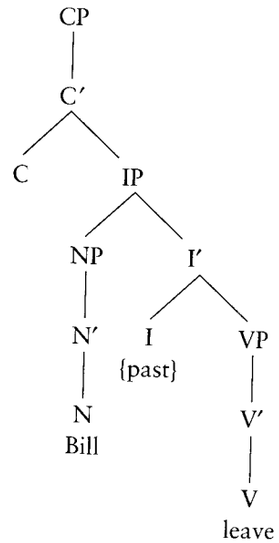
As you can tell by your intuition, the corresponding yes/no question is (9).

(9) Did Bill leave?

Notice that in (8), the verb is marked for past tense, but it isn't in (9). Also, notice that (9) has the word "did," but (8) doesn't. You have probably deduced that these two facts are related.

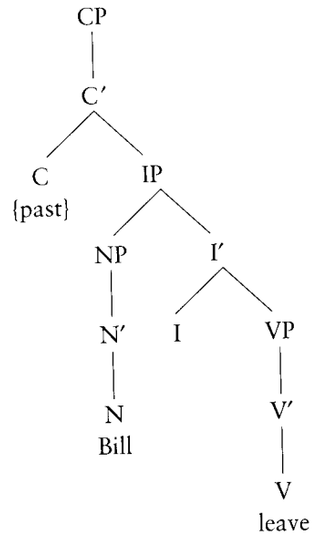
First, consider the phrase marker for (8), given in (10).

(10)



As we saw in chapter 8, in a declarative sentence the inflectional affix hops onto the verb. In this case, {past} joins with "leave" to make "left," as in (8). However, suppose {past} moves to C by I-to-C movement in order to create a yes/no question. This results in the Surface Structure given in (11).

(11)



The restriction on affix-hopping which we noticed in chapter 8 was that the affix must be *adjacent* to the verb. This is true in (10), but not in (11) – in (11) the subject "Bill" comes between {past} and the verb.

In English, we save the structure in (11) by using an operation known as "Do-support." The dummy verb "do" is inserted as a host for the {past} affix in C, and of course "do" + {past} = "did." This gives us the desired surface form (9).

Do-support is only used as a last resort operation. It applies to save a stranded tense affix in questions and in a few other situations. Always apply all other transformations first. If the tense affix ends up next to the verb, then apply affix-hopping. However, if the tense marker is stranded, then apply Do-support.

Do-support in negative sentences

Additional evidence for the operation of Do-support as a last resort comes from negative sentences. Consider the examples in (12)–(13).

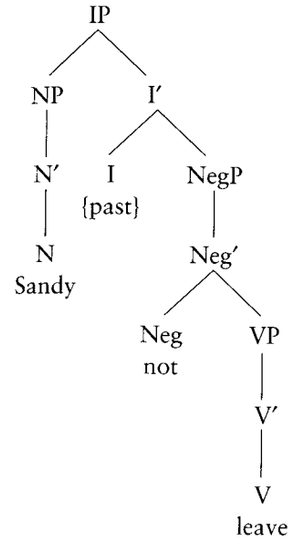
(12) a. Sandy will leave.
b. Sandy will not leave.

(13) a. Sandy left.
b. *Sandy not left.
c. Sandy did not leave.

The examples in (12) show us that a negative sentence looks like a positive sentence with the word "not" before the verb. However, (13) shows that the negative element "not" cannot appear before a tensed verb. Apparently, the tense affix cannot adjoin to the verb by affix-hopping in negative sentences, as (13b) shows. This means the affix will be left stranded. Just as in questions, to save the stranded affix, Do-support must apply.

Let's take a moment to see how this works. First, assume that a negative element such as "not" is the head of a new phrasal category, which we'll call "NegP." NegP follows the X′-schema we saw in chapter 9, and it comes between IP and VP. This means that the structure of (13c) will be that in (14).

(14)



With this tree, it is easy to see that {past} cannot affix-hop onto "leave" because they are not adjacent – the negative "not" intervenes. Thus, Do-support must apply to host the stranded tense affix in I.

Conclusion

Let's step back for a moment, and consider the properties of the mental grammar. It consists of a finite set of PS rules, which generate phrase markers (Deep

Structures), as well as transformational rules which are relatively simple, for converting phrase markers into still other phrase markers. The output of the transformational component is a Surface Structure. Together, the combination of rules is able to generate an infinite array of Surface Structures for a language. Transformational rules are needed in addition to PS rules because sentences of natural language sometimes exhibit dependencies between constituents that are distant from each other (on the surface). These dependencies are characteristic of transformational rules, but not of PS rules.

Bibliographical Comments

We have cited Chomsky's early works in our discussion of Phrase Structure, but it was his development of the transformation which served as a vital new tool for analyzing language. As we have mentioned earlier, it was his reorientation of the goals of linguistics which truly revolutionized the field. These early works, which gave detailed transformational analyses of English, as well as discussions of the limitations of Phrase Structure grammars, include Chomsky (1975) (first distributed in 1955), Chomsky (1957), and Chomsky (1965). The terms Deep Structure and Surface Structure were first introduced in Chomsky (1965), where the notions are discussed in detail. Good introductory textbooks on the transformational approach to grammar described here include Akmajian Demers, Farmer, and Harnish (1995), Baker (1978), Cook and Newson (1996), Radford, (1981), and Riemsdijk and Williams (1986).