MATHEMATICS OF LANGUAGE

Edited by

ALEXIS MANASTER-RAMER
Wayne State University

JOHN BENJAMINS PUBLISHING COMPANY
AMSTERDAM/PHILADELPHIA

1987

ON THE DESIGN OF FINITE TRANSDUCERS FOR PARSING PHRASE-STRUCTURE LANGUAGES*

D. Terence Langendoen
Brooklyn College and
CUNY Graduate Center

Yedidyah Langsam
Brooklyn College and
CUNY Graduate Center

1. INTRODUCTION

There is a long history of research on the design of language production and recognition devices with strictly finite resources that mimic the behavior of those with potentially unbounded memory. The first major achievement in this line of research is the algorithm in Chomsky (1959) for constructing a finite automaton that accepts all and only all the expressions generated by a Chomsky-normal-form context-free phrase-structure grammar with no center embedding. In Langendoen (1961), an attempt was made to strengthen this result by developing a procedure for constructing a finite transducer that not only accepts those expressions, but which associates with each one its structural descriptions with respect to the original phrase-structure grammar. That attempt to build a finite parser for a phrase-structure language, however, was unsuccessful, because the resulting device did not, in fact, have strictly finite memory resources.

In Langendoen (1975), it was shown how Chomsky's original algorithm could be extended to cover arbitrary phrase-structure grammars, not just those in Chomsky-normal form. However, it was also argued that the attempt to design a finite parser for phrase-structure languages is doomed from the outset, because the problem of matching labeled brackets in structural descriptions

* This work was supported in part by grant number 6-64336 from the PSC-CUNY Faculty Research Award Program. We thank Maria Edelstein, David Lawrence, and Dana McDaniel for helpful discussions.

of expressions with multiple left and right embedding requires unbounded memory resources, just as the generation of expressions with multiple center embedding does.

This negative result spurred investigation into ways of modifying labeled bracketing notation so that not every bracket has to be matched and so that the representations of the structural descriptions of noncenter-embedded expressions themselves lack center embedding; see Langendoen (1975, n. 4), Langendoen (1979), Krauwer and des Tombe (1980, 1981), Langendoen and Langsam (1984).

In this paper, we represent the structural descriptions of expressions generated by a context-free phrase-structure grammar GR[1] by means of sequences of statements that express the dominance relations among their constituents, and provide an algorithm for constructing a finite transducer that associates those sequences with those expressions. The performance of this transducer degrades with linguistic complexity much in the manner that a person's does. It is unable to parse correctly any expression that manifests more than second degree center embedding (a limit that can be modified by changing a parameter in the algorithm), and it fails to compute certain parse trees for expressions that combine at least first degree center embedding with greater than second degree right branching. Finally, it provides partial analyses consistent with GR for expressions that are not part of the language generated by GR.

[1]We assume that the productions of GR are all of the form FPR, where *A* is a nonterminal element, *a* is a terminal element, and *X* is a nonnull string of nonterminal elements.

FPR. 1. A --> X
2. A --> a

2. THE TRANSDUCER

Like Krauwer and des Tombe (1980, 1981), we construct a one-way nondeterministic finite transducer with accepting states. The input tape contains a single finite string of symbols over the terminal vocabulary of GR followed by the designated symbol #; the remainder of the input tape is blank. In its initial configuration, the transducer is scanning the first symbol of its input; it is in a designated initial state *I* and the output tape is blank. On any transition, the transducer may read a symbol from the input tape or leave the input tape alone; it may enter a new state; and it may or may not print a string over its output vocabulary (see below) on the output tape. The transducer accepts a string printed on its input tape and associates with that string what it has written on the output tape if and only if the following conditions are jointly satisfied. First, having started in its initial configuration, the transducer is scanning the first blank unit on the input tape to the right of the last symbol on the input tape (the designated symbol #). Second, the transducer has reached the designated final state *F* for the first time.

The form and interpretation of the statements that the transducer writes on the output tape are given in OP, where *A* and *B* are elements of the nonterminal vocabulary of GR and *a* is an element of the terminal vocabulary of GR.

OP(1)	A / a	'a is the first (and only) daughter of A'
(2)	A / B	'B is the first daughter of A'
(3)	A \ B	'B is a nonfirst daughter of A'

For each structural description of an expression with respect to a phrase-structure grammar, there is a sequence of statements of the types in OP that represents it. For example, let GR(1) be a grammar with the nonterminal vocabulary VN(1), the terminal vocabulary VT(1), the axioms AX(1), and the productions PR(1).

VN(1) A, B, C, D, S

VT(1) a, b, c

AX(1) S

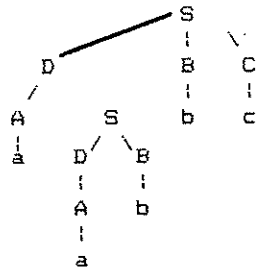
PR(1) a. S --> D B (C)
b. D --> A (S)
c. A --> a
d. B --> b
e. C --> c

Consider the expression EX(1) generated by GR(1).

EX(1) a a b b c

EX(1) has a structural description with respect to GR(1) which is represented by the tree diagram SD(1).

SD(1) Structural description of EX(1) with respect to GR(1).



The sequence of statements in SQ(1) represents the same information as does SD(1), and results from a traversal of that tree in inorder (Langsam, Augenstein and Tenenbaum 1985: 291-2).[2]

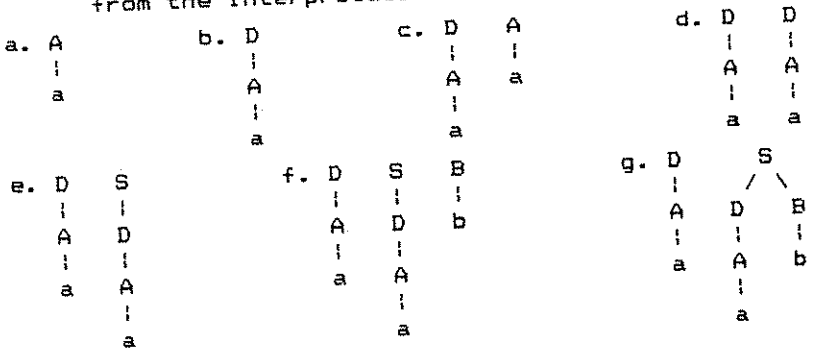
[2]The root is to be visited after its first daughter (if any) is traversed; each subsequent daughter (if any) is then traversed from left to right. Statements of the form *A / a* are to be understood as instructions to traverse the first (and only--see note 1) daughter of *A*, which is

SQ(1) a. A / a
b. D / A
c. A / a
d. D / A
e. S / D
f. B / b
g. S \ B
h. D \ S
i. S / D
j. B / b
k. S \ B
l. C / c
m. S \ C

Our method of interpreting dominance statements as instructions for building tree structures is as follows. Statements of type OP(1) introduce all three elements (mother node, daughter node, and the branch connecting them) as subtrees to the right of all subtrees constructed so far. Statements of type OP(2) introduce the mother node and the branch only, making the mother node the root of the rightmost subtree, whose root was formerly the daughter node. Finally, statements of type OP(3) introduce the branch only, connecting the root of the rightmost subtree (the daughter node) to the root of the subtree immediately to the left (the mother node).

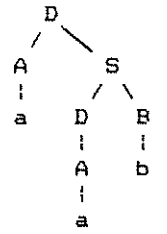
Applying this method of interpretation to the ordered statements in SQ(1), we obtain the results summarized in FG(1).

FG(1) Sequence of partial tree diagrams obtained from the interpretation of DM(1).

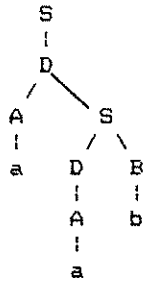


[2]The root is to be visited after its first daughter (if any) is traversed; each subsequent daughter (if any) is then traversed from left to right. Statements of the form *A / a* are to be understood as instructions to traverse the first (and only--see note 1) daughter of *A*, which is *a*, and then to visit the root *A*. The traversal of *a* consists simply of visiting the root *a*, since terminal symbols have no daughters (again, see note 1).

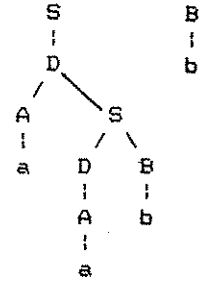
h.



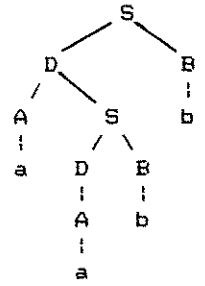
i.



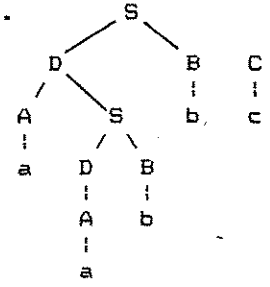
j.



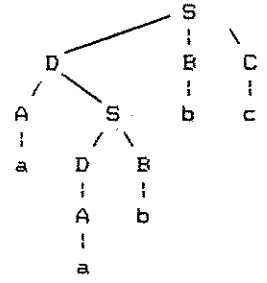
k.



l.



m.



Transitions are represented in the automaton by means of expressions of the form TF.

TF. P --> (w) Q($; D_1(, D_2(, \ldots(, D_k)\ldots)))$

In TF, P and Q are states, w is a symbol on the input tape, and each D_i, $1 \leq i \leq k$, is a statement of dominance relations of the types in OP. The parentheses indicate that the reading in of material from the input tape and the printing of material on the output tape need

not occur on all transitions. The semicolon separates the sequence of output statements from the entered state, and commas separate the output statements from each other.

The states of the transducer include I, F, and strings of the form ST.

ST. $Axy_1 \ldots Axy_n$, where each A is an element of the nonterminal vocabulary of GR, and each x and y are numerical attributes with the following ranges and interpretations:

(1) The x-attribute is either 0 or 1. If $x = 0$, then the associated element is *incomplete*; that is, open to the further attachment of daughters. If $x = 1$, then the associated element is *complete*; that is, closed to the further attachment of daughters. The last element in the state must have the x-attribute of 1.

(2) The y-attribute ranges over the values 0 to 3. If $y > 0$, then the associated element is understood to be a *root* of a subtree, and we use the symbol + for unspecified root values of the y-attribute. These values of y indicate the number of times that the associated elements occur as a root in the state up to that point, starting from the left. The y-attribute can have a value of 3 for at most one element in the state. If $y = 0$, then the associated element is understood to be the rightmost *descendant* of the root to its immediate left in the state. The x-attribute of a descendant must be 0, while

that of the root to its immediate left must be 1. Descendants cannot occur initially or in immediate sequence (since they must always be preceded by a root), nor can they occur finally (since their *x*-attributes must be 0).

From the conditions in ST(2), it follows that the length n of the states of T is linearly bounded by the number of distinct nonterminal symbols in GR. Specifically, if there are p such symbols, then every state of T cannot be longer than $4p+1$ elements, since by ST(2) no state can have more than $2p+1$ root elements and no state can have more than $2p$ descendant elements.

3. THE CONSTRUCTION

In this section, we show how to construct a finite transducer FT that meets the requirements described in section 1 above, and that associates with the expressions it analyzes sequences of dominance relations interpretable as tree structures in the manner described in section 2. FT uses its states to keep track of the sequence of root nodes of the subtrees that appear in the interpretation as the sequence is produced. Alternatively, it could be equipped with a finite auxiliary memory for holding this information. FT handles left embedding without difficulty, but not surprisingly, is unable to handle center embedding beyond a fixed finite degree. However, as matters now stand, it also cannot handle right embedding, the problem being that with right embedding, as with center embedding, the number of subtrees that appear in the interpretation, and consequently the length of the states of the transducer, and hence their number, grow without limit. To solve this problem, FT tracks repetitions of category symbols in its states,

as in ST(2) above. When an element recurs for the third time, all of the elements from the second recurrence to the element immediately preceding the third recurrence are eliminated. We refer to this process as *collapse*, and thanks to it, a fixed, finite bound is placed on the length and hence number of the states of the transducer. Moreover, suppose the element originally to the immediate left of the second recurrence is *A0+* and the element to the immediate left of the third recurrence is *L0+*. Then the former is replaced by *A1+* and the latter by *L00*; that is, the latter is reanalyzed as the rightmost descendant of the former.

We turn now to the construction itself. By definition, FT starts in the designated state *I* reading the first symbol (word) on the input tape. We allow FT to be able to parse any such word which occurs in the terminal vocabulary of GR. That is, it contains all of the transitions in the initial condition CO(1).

CO(1) Initial condition. If GR has the rule:

A --> a

then FT has the transition:

I --> a A11; A / a

In order to terminate in a successful parse, the transducer should be reading the designated symbol # and be in a complete state *B* that corresponds to an axiom of GR.[3] Accordingly, FT has all of the transitions in the final condition CO(2).

[3] Traditionally, generative grammars (more precisely, the syntactic components of such grammars) of natural languages are understood as having only one axiom, usually *S* (for *sentence*). Braine (1979) has argued that each member of the nonterminal vocabulary of the grammar counts as an axiom, on the grounds that grammars should generate natural language expressions of every type. We adopt here the suggestion of Langendoen (1982) that the axioms of a grammar consist of those elements of the nonterminal vocabulary that categorize the phrase types that occur naturally in connected text, minimally *S* and *NP* (for *noun phrase*).

CO(2) Final condition. If *B* is an axiom of GR, then FT has the transition:

B11 --> # F

In the next six conditions, CO(3-8), it is assumed that no root element in a left-hand state has a *y*-attribute of 3 and that *y*-attributes of roots are automatically incremented as necessary in right-hand states. Hence no restrictions are placed on the *y*-attributes of roots in the statement of these conditions. The first such condition concerns the situation in which the root of the rightmost subtree that has been constructed so far occurs as the sole daughter of another constituent.[4]

CO(3) Sole daughter condition. If GR has the rule:

B --> C

then FT has the transitions:

XC1+ --> XB1+; B / C

The next condition is that in which the root of the rightmost subtree that has been constructed so far occurs as a left daughter of another constituent.

CO(4) Left daughter condition. If GR has the rules, where *Z* is nonnull:

a. B --> C Z

b. A --> a

then FT has the transitions:

XC1+ --> a XB0+A1+; B / C, A / a

[4]In this section late letters of the alphabet (*V* to *Z*) are to be understood as variable strings, possibly null unless otherwise specified. The letters *V* to *X* are reserved for strings of nonterminal symbols with associated attributes in states of the transducer FT, and *Y* to *Z* for strings of nonterminal symbols in productions of GR. In the following sections, the symbol *V* is used to stand for a particular nonterminal element.

Next is the condition in which the root of the rightmost subtree becomes a medial daughter of the root of the subtree to the immediate left.[5]

CO(5) Medial daughter condition. If GR has the rules, where *Y* and *Z* are nonnull:

a. B --> Y C Z

b. A --> a

then FT has the transitions:

XB0+C1+ --> a XB0+A1+; B \ C, A / a

Next are two conditions in which the root of the rightmost subtree constructed so far becomes the right daughter of a constituent of the subtree to the immediate left. In the first of these, the mother node is the root of that subtree. In the second case, the mother node is the rightmost descendant of the root of that subtree.

CO(6) First right daughter condition. If GR has the rules, where *Y* is nonnull:

B --> Y C

then FT has the transitions:

XB0+C1+ --> XB1+; B \ C

CO(7) Second right daughter condition. If GR has the rules, where *Y* is nonnull:

B --> Y C

then FT has the transitions:

XB00C1+ --> X; B \ C

Next is the condition in which material other than the end symbol remains to be read in from the input tape,

[5]The medial branching condition CO5 is deliberately formulated in such a way that if GR has a rule of the form *B* --> *E C F*, then FT is constructed as if GR had the schema *B* --> *E C* F*. This formulation is based on the assumption that grammars of natural languages use multiple branching exclusively for coordinate compound structures. If, however, such grammars manifest true ternary (quaternary, etc.) branching, the medial branching condition can be modified to accommodate it.

but the last element of the state cannot be made a daughter of any other constituent. In this case, the next symbol from the input tape is simply read in and analyzed, and no further analysis is done on the previous subtree.

CO(8) No daughter condition. If GR has the rule:

A --> a

and FT is in a state *XC1+*, to which none of the other conditions are applicable, then FT has the transitions:

XC1+ --> XC1+A1+; A / a

The last condition deals with the collapse of states in which an element with a *y*-attribute of 3 appears. In its most general formulation, the collapse condition must be stated to permit a descendant to appear between each root in the chain undergoing collapse. To express the condition in as concise a form as possible, we introduce the convention that corresponding elements appearing in angled brackets must be chosen (i.e., always the first element, or always the second).

CO(9) Collapse condition. If GR has the rules, where *Y* is nonnull and $f \geq 1$:

1. <A --> Y C, B --> Y C>

...

f. <J --> Y L, K --> Y L>

then FT has the transitions:

V<A0+, A1+B00><C02, C12D00>W<J0+,
J1+K00><L0+,L1+M00>C03X -->
VA1+<L00, M00>C02X;
<A, B> \ C, ..., <J, K> \ L

We refer to the sequence *<A0+, A1+B00>...<L0+,L1+M00>* as the *collapse chain*. Note that if $f = 1$, then *W* is null and *L0+ = C02*, *L1+ = C12* and *M00 = D00*.

How the construction works can best be shown by illustrative examples, to which we now turn.

4.1. A TRANSDUCER FOR A FINITE LANGUAGE

Let GR(2) be a phrase-structure grammar with the nonterminal vocabulary VN(2), the terminal vocabulary VT(2), the axioms AX(2) (see n. 3) and the production schemata PR(2).

VN(2) D, N, V, NP, VP, S

VT(2) a. the
b. boy, girl, ..., teacher
c. knew, believed,..., saw

AX(2) NP, S

PR(2) a. S --> NP VP
b. NP --> D N
c. VP --> V (NP)
d. D --> the
e. N --> {boy, girl, ..., teacher}
f. V --> {knew, believed, ..., saw}

GR(2) generates the finite set of expressions in LG(2).

LG(2) a. VT(2)a VT(2)b
b. LG(2)a VT(2)c (LG(2)a)

Finally, let IL(2) consist of all strings of LG(2) followed by the end marker #, and let UL(2) consist of all strings over VT(2) followed by #. We now use the construction given in section 3 to form a finite transducer FT(2) that accepts the members of IL(2) and associates with them their structural descriptions with respect to GR(2); and that assigns partial structural des-

criptions to all other strings in UL(2). FT(2) has the transitions represented in the schemata in TS(2)a-f; to the right are given the conditions of the construction which license the schemata. In these schemata, *L* ranges over the lexical categories in GR(2) (*D*, *N* and *V*), and *w* ranges over VT(2), such that *L* --> *w* is a member of PR(2).

TS(2) a. I --> w L11; L/w CO(1)

b. 1. NP11 --> # F CO(2)
2. S11 --> # F

c. XV1+ --> XVP1+; VP/V CO(3)

d. 1. XD1+ --> w XNP0+L1+; NP/D, L/w CO(4)
2. XNP1+ --> w XS0+L1+; S/NP, L/w
3. XV1+ --> w VP0+L1+; VP/V, L/w

e. 1. XS0+VP1+ --> XS1+; S\VP CO(6)
2. XNP0+N1+ --> XNP1+; NP\N
3. XVP0+NP1+ --> XVP1+; VP\NP

f. 1. XN1+ --> w XN1+L1+; L/w CO(8)
2. XVP1+ --> w XVP1+L1+; L/w
3. XS1+ --> w XS1+L1+; L/w

We illustrate the operation of the transducer FT(2) first by showing in FG(2)a how it analyzes the input string EX(2)a, which is a member of IL(2).

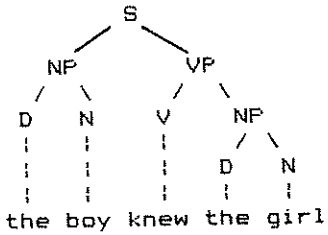
EX(2) a. the boy knew the girl #

FG(2) a. Analysis of EX(2)a by FT(2).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	boy	NP01N11	NP/D, N/boy	TS(2)d1
3.		NP11	NP\N	TS(2)e2
4.	knew	S01V11	S/NP, V/knew	TS(2)d2
5.	the	S01VP01D11	VP/V, D/the	TS(2)d3
6.	girl	S01VP01NP01N11	NP/D, N/girl	TS(2)d1
7.		S01VP01NP11	NP\N	TS(2)e2
8.		S01VP11	VP\NP	TS(2)e3
9.		S11	S\VP	TS(2)e1
10.	#	F		TS(2)b2

It may be readily verified that the sequence of statements that appears on the output tape of FT(2) upon processing EX(2)a is equivalent to the tree diagram in SD(2)a, according to the interpretive rules stated in section 2.

SD(2) a. Tree diagram associated with EX(2)a by FT(2).



Clearly SD(2)a represents the structural description of EX(2)a with respect to GR(2).

We next illustrate the operation of FT(2) by showing how it is capable of analyzing three strings in UL(2) which are not in IL(2); i.e., which are ungrammatical with respect to GR(2). We begin with the string in EX(2)b.

EX(2) b. the knew boy #

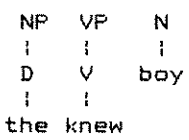
In FG(2)b, we show one way in which FT(2) is able to assign a partial analysis to this string.

FG(2) b. Possible analysis of EX(2)b by FT(2).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	knew	NP01V11	NP/D, V/knew	TS(2)d1
3.	boy	NP01VP01N11	VP/V, N/boy	TS(2)d3
4.	(Reading #; no further transitions are possible.)			

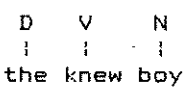
The tree diagram that is associated with the sequence of output statements in FIG(2)b is given in SD(2)b.

SD(2) b. Tree diagram associated with EX(2)b by FT(2).



In SD(2)b, the root symbols NP and VP correspond to incomplete elements in the last state that the transducer reaches before it can proceed no further, and can be thought of as uncorroborated guesses that the transducer has made in attempting to process EX(2)b. Consequently, the transducer has established that the string EX(2)b can in fact be analyzed as a sequence of three words with the categorization shown in SD(2)b1.

SD(2) b. 1. Actual analysis of EX(2)b by FT(2) in FG(2)b.



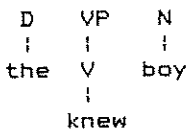
FT(2) is also able to give a slightly different analysis of the string in EX(2)b, in virtue of the fact that it can analyze in step 3 the word *knew* as a complete *VP*. If it does so, the analysis would continue as in FG(2)b2.

FG(2) b. 2. Another possible analysis of EX(2)b by FT(2).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1-2.		(as in FG(2)b)		
3.		NP01VP11	VP/V	TS(2)c
4.	boy	NP01VP11N11	N/boy	TS(2)f1
5.	(Reading #; no further transitions are possible.)			

As a result of this sequence of steps, the transducer associates with EX(2)b the actual analysis shown in SD(2)b2.

SD(2) b. 2. Another analysis of EX(2)b by FT(2).



The success of this analysis depends on the use of a transition licensed by CO8, which enables a new subtree to be constructed to the right of a subtree whose root is incomplete. The use of such a transition is also required in any analysis by FT(2) of EX(2)c, the second ungrammatical string we consider.

EX(2) c. knew the boy the girl #

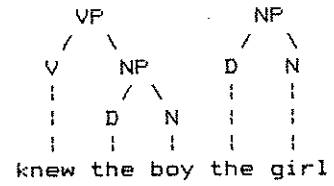
A possible analysis of EX(2)c by FT(2) is given in FG(2)c; another one, in which *knew* is analyzed as *VP*, is ignored here.

FG(2) c. Possible analysis of EX(2)c by FT(2).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	knew	V11	V/knew	TS(2)a
2.	the	VP01D11	VP/V, D/the	TS(2)d3
3.	boy	VP01NP01N11	NP/D, N/boy	TS(2)d1
4.		VP01NP11	NP\N	TS(2)e2
5.		VP11	VP\NP	TS(2)e3
6.	the	VP11D11	VP/V, D/the	TS(2)f2
7.	girl	VP11NP01N11	NP/D, N/girl	TS(2)d1
8.		VP11NP11	NP\N	TS(2)e2
9.	(Reading #; no further transitions are possible.)			

The sequence of statements on the output tape in FG(2)c is equivalent to the forest of trees in SD(2)c.

SD(2)c. Tree diagram associated with EX(2)c by FT(2).



In this case, since the elements *VP* and *NP* in the last state that the transducer reaches in FG(2)c are both specified as complete, the structure shown in SD(2)c is exactly the one which the transducer associates with the string EX(2)c.

The ability of FT(2) to provide partial analyses of arbitrary sequences of words drawn from the terminal vocabulary of GR(2) is limited only by the length and number of its states, as we see from the analysis of EX(2)d, the third ungrammatical string we consider.

EX(2) d. boy girl teacher student #

FG(2) d. Attempted analysis of EX(2)d by FT(2).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	boy	N11	N/boy	TS(2)a
2.	girl	N11N12	N/girl	TS(2)f1
3.	teacher	N11N12N13	N/teacher	TS(2)f1
4.	(Reading *student*; no further transitions are possible.)			

After step 3 in FG(2)d, only a transition licensed by the collapse condition CO9 would be applicable, since the transducer is in a state containing an element with a *y*-attribute of 3. However, there are no such transitions, for two reasons. First, the elements at the

beginning and the end of the collapse chain, *N11* and *N12*, are both complete, rather than incomplete, as required by the collapse condition. Second, GR(2) does not contain rules in which the elements in the collapse chain are introduced as right daughters.

It would be desirable to enable the transducer to complete the processing of strings like EX(2)d; hence we relax the requirements on the collapse condition as in CO(9R).

CO(9R) Relaxation of the collapse condition CO(9). If *CO2*, *CO3* and the roots immediately preceding them are complete instead of incomplete as required by CO9, then FT has transitions in which the elements of the collapse chain are not connected as right daughters and the *y*-attribute of the last element of the chain is not changed to 0.

As a result of CO(9R), TR(2) contains transitions such as the one in TS(2)g, which allow it to continue the processing of examples like EX(2)d.*

TS(2) g. N11N12N13 --> N11N12 CO(9R)

The completion of the analysis of EX(2)d is shown in FG(2)d1.

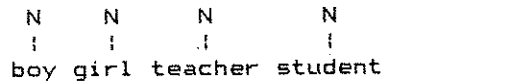
FG(2) d. 1. Completion of analysis of EX(2)d by FT(2).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1-3.		(as in FG(2)d)		
4.		N11N12		TS(2)g
5.	student	N11N12N13	N/student	TS(2)f1
6.		N11N12		TS(2)g
7.	(Reading #; no further transitions are possible.)			

*CO(9R), in fact, allows for the construction of a large number of distinct transitions in TR(2). The problem of specifying them exactly is a difficult one, and we do not pursue it further here.

The output, in this case, is equivalent to a forest of trees analyzing the words in EX(2)d as instances of the category *N*, as shown in SD(2)d.

SD(2) d. Analysis of EX(2)d by TR(2).



4.2. A TRANSDUCER FOR A LANGUAGE WITH LEFT-EMBEDDING

Next, let GR(3) be an extension of the grammar GR(2) in section 4.1 with the additional elements as follows:

VN(3) G, GP

VT(3) 's

PR(3) a. NP --> GP N

b. GP --> NP G

c. G --> 's

Henceforth, the sets VN(3), VT(3), AX(3) and PR(3), are assumed to include all of the members of the corresponding sets VN(2), VT(2), AX(2) and PR(2). GR(3) generates the infinite set of expressions in LG(3).

LG(3) a. VT(2)a (VT(2)b VT(3))* VT(2)b

b. LG(3)a VT(2)c (LG(3)a)

The set IL(3) is formed from EX(3) by the addition of the end marker #, analogous to IL(2); and the set UL(3) is any string over VT(3) followed by #, analogous to UL(2).

Given this extension of the grammar, we construct a transducer FT(3), which contains all of the transitions schematized in TS(2), together with those in TS(3).[7]

[7]To simplify the remaining discussion, we consider only the problem of constructing transducers that associate structural descriptions with expressions that are generated by particular grammars; hence we omit consideration of additional transition schemata that are needed to analyze strings that are not generated by those grammars.

TS(3) a. 1. XNP1+ --> 's XGP0+G1+; CO(4)
GP/NP, G/'s

2. XGP1+ --> VT(2)b XNP0+N1+;
NP/GP, N/VT(2)b

b. XGP0+G1+ --> XGP1+; GP\G CO(6)

In FG(3)a, we show how FT(3) analyzes the input string EX(3)a.

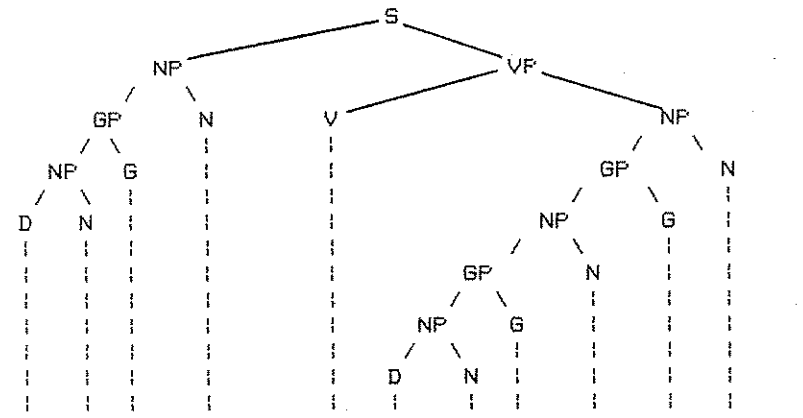
EX(3) a. the boy's teacher knew the girl's cousin's friend #

FG(3)a. Analysis of EX(3)a by T(3).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	boy	NP01N11	NP/D, N/boy	TS(2)d1
3.		NP11	NP\N	TS(2)e2
4.	's	GP01G11	GP/NP, G/'s	TS(3)a1
5.		GP11	GP\G	TS(3)b
6.	teacher	NP01N11	NP/GP, N/teacher	TS(3)a2
7.		NP11	NP\N	TS(2)e2
8.	knew	S01V11	S/NP, V/knew	TS(2)d2
9.	the	S01VP01D11	VP/V, D/the	TS(2)d3
10.	girl	S01VP01NP01N11	NP/D, N/girl	TS(2)d1
11.		S01VP01NP11	NP\N	TS(2)e2
12.	's	S01VP01GP01G11	GP/NP, G/'s	TS(3)a1
13.		S01VP01GP11	GP\G	TS(3)b
14.	cousin	S01VP01NP01N11	NP/GP, N/cousin	TS(3)a2
15.		S01VP01NP11	NP\N	TS(2)e2
16.	's	S01VP01GP01G11	GP/NP, G/'s	TS(3)a1
17.		SVPGP	GP\G	TS(3)b
18.	friend	S01VP01NP01N11	NP/GP, N/friend	TS(3)a2
19.		S01VP01NP11	NP\N	TS(2)e2
20.		S01VP11	VP\NP	TS(2)e3
21.		S11	S\VP	TS(2)e1
22.	#	F		TS(2)b2

SD(3)a displays the tree diagram that FT(3) associates with EX(3)a.

SD(3) a. Tree diagram associated with EX(3)a by FT(3).



4.3. A TRANSDUCER FOR A LANGUAGE WITH RIGHT-EMBEDDING · Next, let GR(4) be an extension of GR(3) with the same vocabulary and axioms, and with the one additional production in PR(4).

PR(4) VP --> V S

GR(4) generates the infinite set of expressions in LG(4).

LG(4) a. LG(3)a

b. (LG(4)a VT(2)c)* LG(4)a VT(2)c (LG(4)a)

IL(4) is formed from LG(4), as before, by the addition of the end marker #, and UL(4) is identical to UL(3), since no new terminal symbols have been added in this extension.

Given this extension, we construct a transducer FT(4), which contains all of the transitions schematized in TS(2-3), together with the following.

TS(4) a. XVP0+S1+ --> XVP1+; VP\S CO(6)

b. XVP00S1+ --> X; VP\S CO(7)

c. WVP0yS02VP0+S03X --> WVP1+VP00S02X; VP\S, S\VP CO(9)

In FG(4)a, we show how FT(4) analyzes the input string EX(4)a, which manifests first-degree right branching with respect to GR(4).

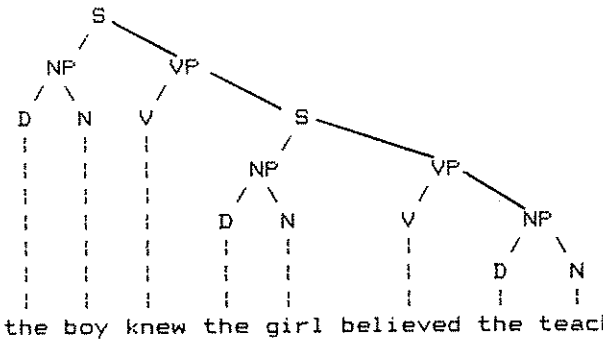
EX(4) a. the boy knew the girl believed the teacher #

FG(4) a. Analysis of EX(4)a by TR(4).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	boy	NP01N11	NP/D, N/boy	TS(2)d1
3.		NP11	NP\N	TS(2)e2
4.	knew	S01V11	S/NP, V/knew	TS(2)d2
5.	the	S01VP01D11	VP/V, D/the	TS(2)d3
6.	girl	S01VP01NP01N11	NP/D, N/girl	TS(2)d1
7.		S01VP01NP11	NP\N	TS(2)e2
8.	believed	S01VP01S02V11	S/NP, V/believed	TS(2)d2
9.	the	S01VP01S02VP02D11	VP/V, D/the	TS(2)d3
10.	teacher	S01VP01S02VP02NP01N11	VP/D, N/teacher	TS(2)d1
11.		S01VP01S02VP02NP11	NP\N	TS(2)e2
12.		S01VP01S02VP12	VP\NP	TS(2)e3
13.		S01VP01S12	S\VP	TS(2)e1
14.		S01VP11	VP\S	TS(4)a
15.		S11	S\VP	TS(2)e1
16.	#	F		TS(2)b2

The tree diagram that is associated with the sequence of output statements in FG(4)a is shown in SD(4)a.

SD(4) a. Tree diagram associated with EX(4)a by TR(4).



After step 7, FT(4) could have followed a different path; if instead of applying a transition based on TS(2)d2, it were to apply one based on TS(2)e3, making the second *NP* a daughter of the first *VP*, then it would analyze EX(4)a as a sequence made of an *S (the boy knew the girl)* and a *VP (believed the teacher)*. The transducer can be designed to avoid this alternative analysis by causing it to favor a transition which delays the closing of a constituent (in this case *VP*) to one which does not, a strategy widely known as *late closure* (Kimball 1974).

Comparing FG(4)a with FG(3)a, it will be noted that right branching, unlike left branching, results in the repetition of nonterminal symbols in the states of the transducer. Clearly, if such repetition were allowed to continue without limit, the device that results from the construction would not be a finite transducer. The collapse condition limits the number of repetitions of nonterminal symbols by collapsing states with three occurrences of a given nonterminal element to states with two such occurrences. The transition schema TS(4)c is based on this condition; an application of one of its rules is

illustrated in FG(4)b, which traces the steps by which FT(4) analyzes the string EX(4)b, which manifests second-degree right branching.

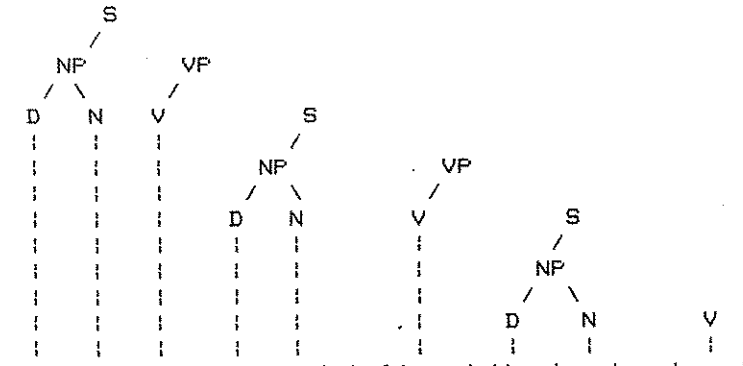
EX(4) b. the boy knew the girl believed the teacher heard the student #

FG(4)b. Analysis of EX(4)b by TR(4).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1-11.		(as in FG(4)a)		
12.	heard	S01VP01S02VP02S03V11	S/NP, V/heard	TS(2)d2
13.		S01VP01VP00S02V11	VP\S, S\VP	TS(4)c
14.	the	S01VP01VP00S02VP02D11	VP/V, D/the	TS(2)d3
15.	student	S01VP01VP00S02VP02NP01N11	NP/D, N/student	TS(2)d1
16.		S01VP01VP00S02VP02NP11	NP\N	TS(2)e2
17.		S01VP01VP00S02VP12	VP\NP	TS(2)e3
18.		S01VP01VP00S12	S\VP	TS(2)e1
19.		S01VP01	VP\S	TS(4)b
20.		S11	S\VP	TS(2)e1
21.	#	F		TS(2)b2

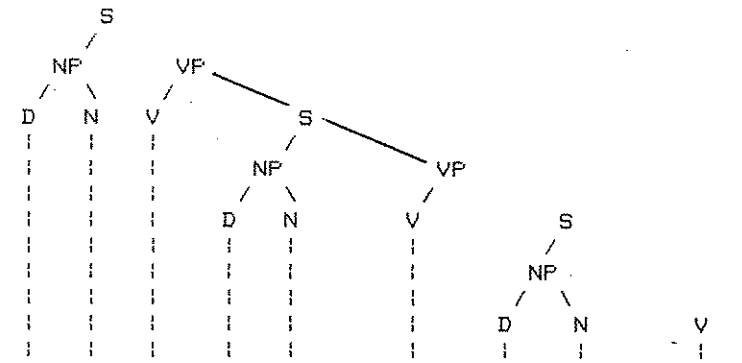
After step 12 in FG(4)b, the output sequence is equivalent to the structure shown in SD(4)b1.

SD(4) b. 1. Interpretation of output in FG(4)b after step 12.



After step 13, upon application of a transition based on TS(4)c, the output is equivalent to that shown in SD(4)b2.

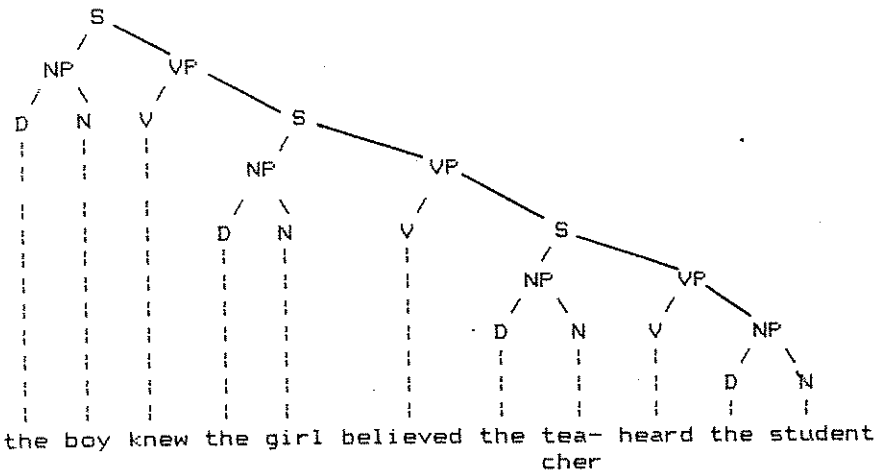
SD(4) b. 2. Interpretation of output in FG(4)b after step 13.



At this point, FT(4) is able to complete the parse of EX(4)b in essentially the same manner in which it completes that of EX(4)a, starting with step 8 of FG(4)a. However, at step 19 in FG(4)b, which corresponds to step 14 in FG(4)a, the bottommost *S* (which at this point is represented in the state of the transducer with a *y*-attribute of 2) is understood as attaching to the middle *VP*, (which at this point is represented in the state with a *y*-attribute of 0, i.e., a descendant), not to the topmost *VP*. Since the middle *VP* is already attached, its representing element disappears from the state along with that of the *S* which becomes its daughter.

In SD(4)b3, we provide a tree diagram of the complete structural description that FT(4) assigns to EX(4)b.

SD(4) b. 3. Interpretation of the complete output in FG(4)b.



Expressions with greater than second degree right branching, such as EX(4)c, are handled similarly, as FG(4)c shows.

EX(4) c. the boy knew the girl believed the teacher heard the student noticed the doctor #

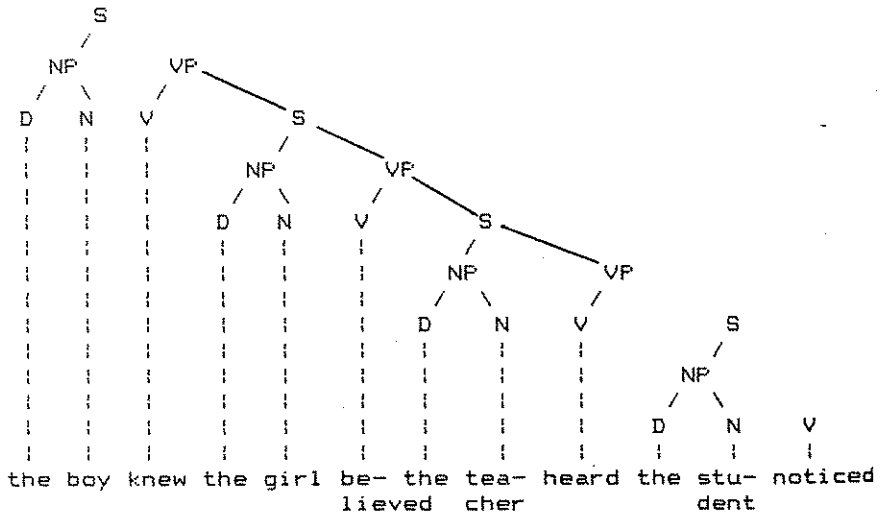
FG(4)c. Analysis of EX(4)c by FT(4).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1-16.		(as in FG(4)b)		
17.	noticed	S01VP01VP00S02VP02S03V11	S/NP, V/noticed	TS(2)d2
18.		S01VP01VP00S02V11	VP\S, S\VP	TS(4)c
19.	the	S01VP01VP00S02VP01D11	VP/V, D/the	TS(2)d3
20.	doctor	S01VP01VP00S02VP01NP01N11	NP/D, N/doctor	TS(2)d1
21-26.		(same as steps 16-21 in FG(4)b)		

In this analysis, two transitions based on the collapse condition are made, first at step 13 (cf. FG(4)b), and again at step 18. Comparing the states immediately prior to collapse (in steps 12 and 17), it will be observed that they differ in that the second one has a descendant

element (*VP00*) immediately preceding *S02* which the first one lacks. Consequently, in step 18, the node represented by *S02* is made a right daughter of the descendant node, whereas in step 13 it is made a right daughter of the root node represented by *VP01*. In both cases, the node formerly represented by *VP02* becomes represented by *VP00*. The result of the operation in step 18 of FG(4)c is shown in SD(4)c.

SD(4) c. Interpretation of output in FG(4)c after step 18.



In like manner, FT(4) is able to parse expressions of IL(4) with any degree of right branching whatever.

4.4. A TRANSDUCER FOR A LANGUAGE WITH CENTER EMBEDDING

Next, let GR(5) be an extension of GR(4) with the following additional elements.

VN(5) AV

VT(5) recently

PR(5) a. S --> NP VP AV

b. AV --> recently

GR(5) generates the set of expressions in LG(5).

LG(5) a. LG(4)a

b. (LG(5)a VT(2)c)^m LG(5)a VT(2)c (LG(5)a (VT(5))^n, $m \geq n$

The sets IL(5) and UL(5) are defined as before.

The transducer FT(5) contains all of the transitions schematized in TS(2-3), together with the following.

TS(5) a. XS0+VP1+ --> recently XS0+AV1+; S\VP, AV/recently CO(5)

b. XS0+AV1+ --> XS1+; S\AV CO(6)

In FG(5)a, we show how FT(5) analyzes the member of IL(5) given in EX(5)a.

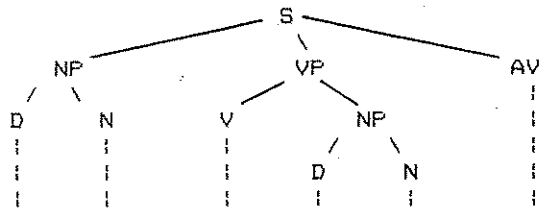
EX(5) a. the teacher noticed the student recently #

FG(5) a. Analysis of EX(5)a by FT(5).

STEP	READ-IN	IN-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	teacher	NP01N11	NP/D, N/boy	TS(2)d1
3.		NP11	NP\N	TS(2)e2
4.	noticed	S01V11	S/NP, V/knew	TS(2)d2
5.	the	S01VP01D11	VP/V, D/the	TS(2)d3
6.	student	S01VP01NP01N11	NP/D, N/girl	TS(2)d1
7.		S01VP01NP11	NP\N	TS(2)e2
8.		S01VP11	VP\NP	TS(2)e3
9.	recently	S01AV11	S\VP, AV/recently	TS(5)a
10.		S11	S\AV	TS(5)b
11.	#	F		TS(2)b2

The sequence of output statements in FG(5)a is equivalent to the tree diagram in SD(5)a.

SD(5) a. Analysis of EX(5)a by FT(5).

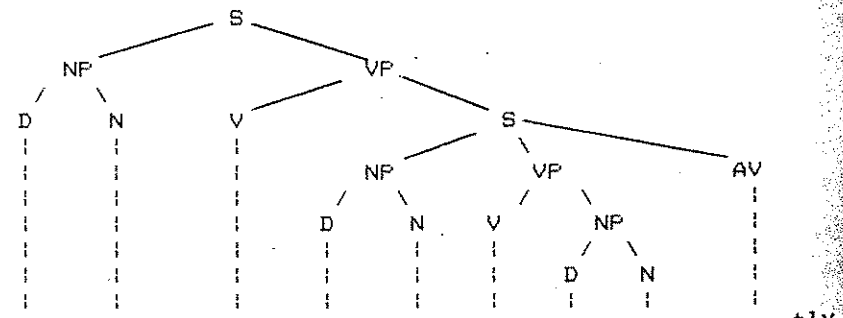


GR(5) allows an *AV* to be a constituent of any *S* in an expression of LG(5); if more than one *AV* occurs, then the expression manifests center embedding. But even if only one *AV* occurs, first degree center embedding results if the expression contains at least one subordinate *S*. Such expressions are also ambiguous, the ambiguity having to do with which *S* the *AV* is associated with. We consider two examples, starting with EX(5)b.

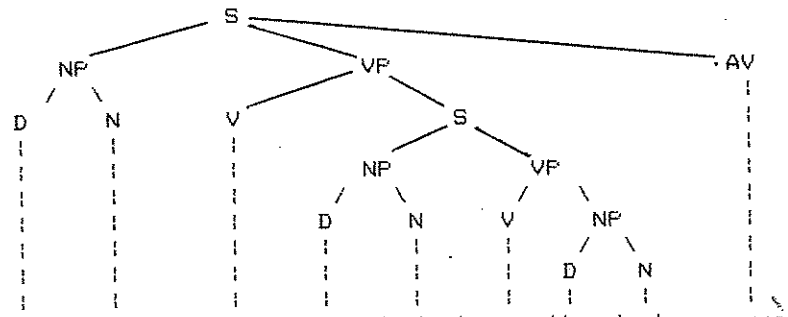
EX(5) b. the teacher noticed the student saw the doctor recently #

The two structural descriptions of EX(5)b with respect to GR(5) are diagrammed in SD(5)b1-2.

SD(5) b. 1. One structural description of EX(5)a.



SD(5) b. 2. Another structural description of EX(5)a.



SD(5)b1 manifests first degree center embedding of the element *VP*; SD(5)b2 manifests first degree center embedding of the element *S*.

FT(5) is able to associate both structural descriptions with EX(5)b, as FG(5)b1-2 show.

FG(5) b. 1. First analysis of EX(5)b by FT(5).

STEP	READ-IN	IN-STATE	OUTPUT	RULE-FROM
1-7.		(same as in FG(5)a)		
8.	saw	S01VP01S02V11	S/NP, V/saw	TS(2)d2
9.	the	S01VP01S02VP02D11	VP/V, D/the	TS(2)d3
10.	doctor	S01VP01S02VP02NP01N11	NP/D, N/doctor	TS(2)d1
11.		S01VP01S02VP02NP11	NP\N	TS(2)e2
12.		S01VP01S02VP12	VP\NP	TS(2)e3
13.	recently	S01VP01S02AV11	S\VP, AV/recently	TS(5)a
14.		S01VP01S12	S\AV	TS(5)b
15.		S01VP11	VP\S	TS(3)a
16.		S11	S\VP	TS(2)e1
17.	#	F		TS(2)b2

FG(5) b. 2. Second analysis of EX(5)b by FT(5).

STEP	READ-IN	IN-STATE	OUTPUT	RULE-FROM
1-12.		(same as in FG(5)b1)		
13.		S01VP01S12	S\VP	TS(2)e1
14.		S01VP11	VP\S	TS(4)a
15.	recently	S01AV11	S\VP, AV/recently	TS(5)a
16.		S11	S\AV	TS(5)b
17.	#	F		TS(2)b2

The choice between these two analyses is made at step 13, where it is decided whether to read in the next word *recently* from the input tape and analyze its mother node *AV* the daughter of the subordinate *S*, or to close the subordinate *S* before reading in the next word. Native speakers of English show a well-known bias toward the first of these two interpretations (Kimball 1974). We can account for this bias in the same way that we accounted for the preference English speakers have to understand EX(4)a as a single complex *S* rather than as a sequence made up of a simple *S* followed by a *VP*; namely by building a preference for late closure into FT(5). A slightly different explanation for biases like those shown toward EX(5)b would be needed, however, if we assumed, as many linguists do, that the *AV* constituent is not introduced simultaneously as a daughter of *S* and sister of *VP*, but rather as a sister of *VP* and a daughter of an intermediate category, call it *VP'*, as in the productions schematized in PR(5').

PR(5') a. S --> NP VP'
b. VP' --> VP (AV)

The corresponding transducer FT(5') would then have to have built into it a preference for attaching an optional constituent as a right sister to the last element in the state if that element is a possible left sister of that constituent, rather than failing to attach

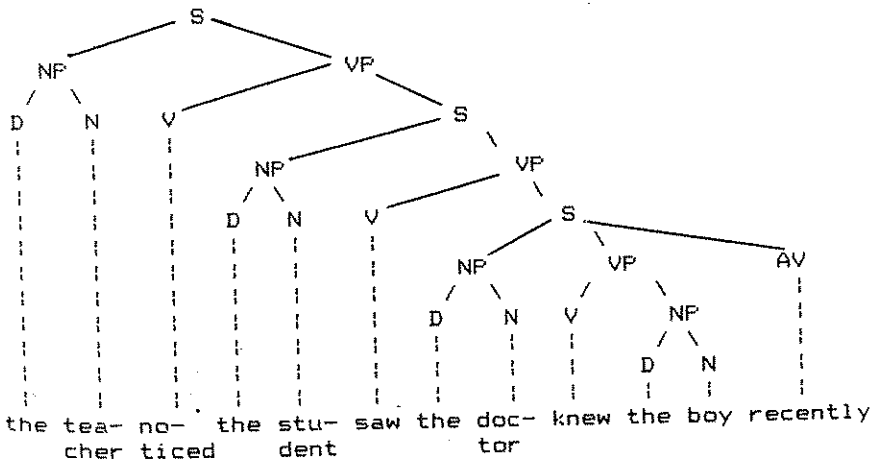
that constituent and creating a closed single-branching mother node for the last element in the state. Such a preference, which Kimball (1974) correctly observes is closely related to, but distinct from, the preference for late closure, he calls *right association*.

Now consider EX(5)c, the second of our ambiguous examples.

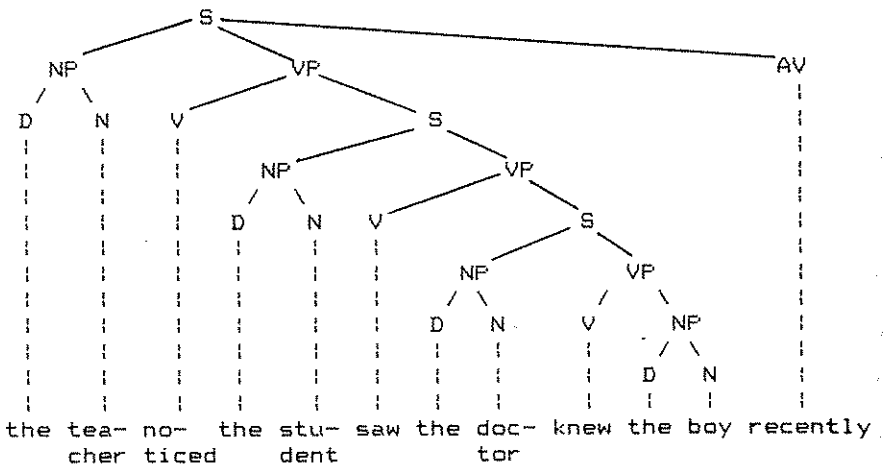
EX(5) c. the teacher noticed the student saw the doctor knew the boy recently #

EX(5)b has three possible interpretations, depending on whether the *AV* constituent is the daughter of the bottom, top, or middle *S* in the structural description, as shown in SD(5)c1-3.

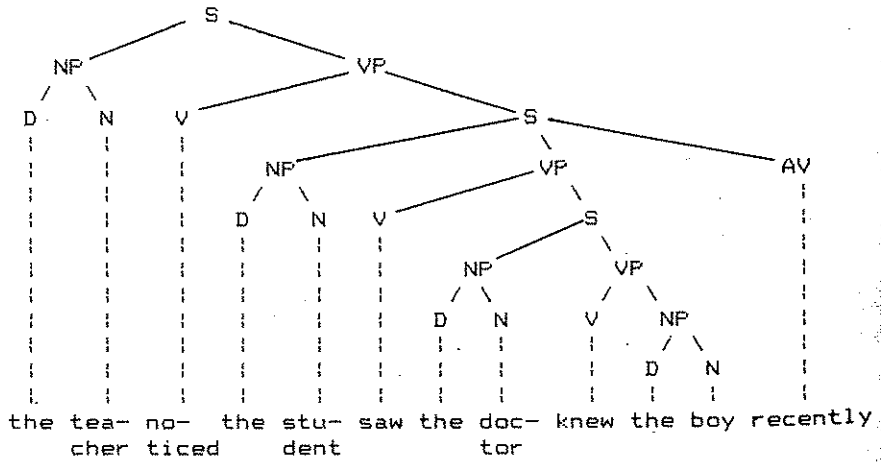
SD(5) c. 1. First structural description of EX(5)c.



SD(5) c. 2. Second structural description of EX(5)c.



SD(5) c. 3. Third structural description of EX(5)c.



Native speakers show by far the strongest preference for the interpretation of EX(5)c corresponding to SD(5)c1, in which the *AV* constituent is part of the bottom *S*; of the remaining two, they again strongly prefer the interpretation corresponding to SD(5)c2, in

which the *AV* constituent is part of the top *S*. They find it extremely difficult to accept the interpretation in which the *AV* constituent is part of the middle *S*, as in SD(5)b3. Such patterns of preference were first pointed out by Gordon (1982).

We account for the preference for the interpretation corresponding to SD(5)c1 over that corresponding to SD(5)c2 by invoking the strategy of late closure (or right association, if the alternative grammar for introducing *AV* constituents is adopted). We account for the preference of both of these interpretations to that corresponding to SD(5)c3 by means of the collapse condition. To see this, consider FG(5)c1-2, which show the two ways in which FT(5) can analyze EX(5)c as an expression of IL(5).

FG(5) c. 1. First analysis of EX(5)c by FT(5).

STEP	READ-IN	IN-STATE	OUTPUT	RULE-FROM
1-11.		(same as in FG(5)b2)		
12.	knew	S01VP01S02VP02S03V11		
			S/NP, V/knew	TS(2)d2
13.		S01VP01VP00S02V11	VP\S, S\VP	TS(4)c
14.	the	S01VP01VP00S02VP02D11		
			VP/V, D/the	TS(2)d3
15.	boy	S01VP01VP00S02VP02NP01N11		
			NP/D, N/boy	TS(2)d1
16.		S01VP01VP00S02VP02NP11		
			NP\N	TS(2)e2
17.		S01VP01VP00S02VP12	VP\NP	TS(2)e3
18.	recently	S01VP01VP00S02AV11	S\VP, AV/recently	TS(5)a
19.		S01VP01VP00S12	S\AV	TS(5)b
20.		S01VP11	VP\S	TS(4)b
21.		S11	S\VP	TS(2)e1
22.	#	F		TS(2)b2

FG(5) c. 2. Second analysis of EX(5)c by FT(5).

STEP	READ-IN	IN-STATE	OUTPUT	RULE-FROM
1-17.		(same as in	FG(5)c1)	
18.		S01VP01VP00S12	S\VP	TS(2)e1
19.		S01VP11	VP\S	TS(4)b
20.	recently	S01AV11	S\VP, AV/recently	TS(5)a
21.		S11	S\AV	TS(5)b
22.	#	F		TS(2)b2

The outputs of FG(5)c1-2 are equivalent to the structures in SD(5)c1-2, respectively. However, there is no way in which FT(5) can analyze EX(5)c as SD(5)c3, since at the point at which FT(5) is ready to read in the word *recently* from the input tape, the element corresponding to the medial *S* node has been deleted from the state by a transition licensed by the collapse condition and hence is not available for the attachment of the *AV* node.

4.5. MORE ON CENTER EMBEDDING

Our final illustration concerns the well studied case of center embedding in English involving relative clause modifiers of noun phrases as in EX(6)a-c.

EX(6) a. the boy the girl knew #

b. the boy the girl the teacher noticed knew #

c. the boy the girl the teacher the doctor saw noticed knew #

These examples exhibit increasing degrees of center embedding of noun phrases (from 1 to 3) and of relative clauses (from 0 to 2).

To generate expressions such as these (without the end markers), let GR(6) be an extension of GR(5) with the following additional elements.

VN(6) RS, RV

PR(6) a. NP --> D N RS

b. RS --> NP RV

c. RV --> V

The categories *RS* and *RV* can be thought of as 'relativized' counterparts to the categories *S* and *VP*.

GR(6) generates the set of expressions in LG(6).

LG(6) a. $(LG(5)a)^k\ (VT(2)c)^{k-1},\ k > 0$

b. $(LG(6)a\ VT(2)c)^m\ LG(6)a\ VT(2)c\ (LG(6)a)\ (VT2d1)^n,\ m \geq n$

The sets IL(6) and UL(6) are defined as before.

The transducer FT(6) contains all of the transition schemata in TS(2-5), together with the following.

TS(6) a. XV1+ --> XRV1+; RV/V CO(3)

b. XNP1+ --> VT(2)c XRS01V11;
RS/NP, V/VT(2)c CO(4)

c. XNP0+N11 --> the XNP0+D11;
NP\N, D/the CO(5)

d. 1. XNP0+RS1+ --> XNP1+; NP\RS CO(6)

2. XRS0+RV1+ --> XRS1+; RS\RV

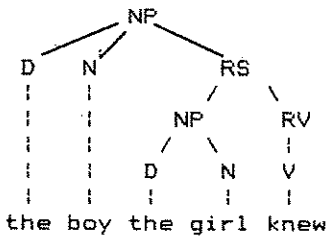
In FG(6)a, we show how FT(6) is able to analyze EX(6)a.

FG(6) a. Analysis of EX(6)a by FT(6).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	boy	NP01N11	NP/D, N/boy	TS(2)d1
3.	the	NP01D11	NP\N, D/the	TS(6)c
4.	girl	NP01NP02N11	NP/D, N/girl	TS(2)d1
5.		NP01NP12	NP\N	TS(2)e2
6.	knew	NP01RS01V11	RS/NP, V/knew	TS(6)b
7.		NP01RS01RV11	RV/V	TR(6)a
8.		NP01RS11	RS\RV	TR(6)d2
9.		NP11	NP\RS	TR(6)d1
10.	#	F		TR(2)b1

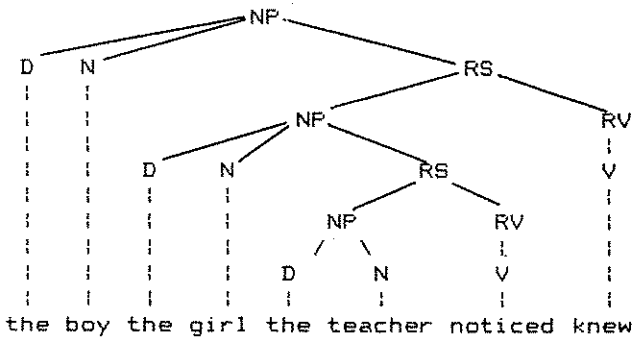
The output in FG(6)a is equivalent to the tree diagram in SD(6)a.

SD(6) a. Structural description of EX(6)a with respect to GR(6).



The structural description of EX(6)b with respect to GR(6) is shown in SD(6)b1.

SD(6) b. 1. Structural description of EX(6)b with respect to GR(6).



When it attempts to analyze EX(6)b, however, FS(6) is stymied by entering a state containing an element with a *y*-attribute of 3 from which neither the original nor the relaxed version of the collapse condition legitimates an exit, as FG(6)b1 shows.

FG(6) b. 1. Initial analysis of EX(6)b by FT(6).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
1.	the	D11	D/the	TS(2)a
2.	boy	NP01N11	NP/D, N/boy	TS(2)d1
3.	the	NP01D11	NP\N, D/the	TS(6)c
4.	girl	NP01NP02N11	NP/D, N/girl	TS(2)d1
5.	the	NP01NP02D11	NP\N, D/the	TS(6)b1
6.	teacher	NP01NP02NP03N11	NP/D, N/teacher	TS(6)b1

At this point, for the transducer to be able to continue, it would need a transition based on CO9 of the form TR(6)e. (No transition based on CO9R is relevant, since the elements in the chain to be collapsed are all incomplete.)

TS(6) e. NP01NP02NP03N11 --> NP01NP00NP03N11; NP\NP

However, such a transition is not available, since there is no rule in GR(6) of the form *NP --> Y NP*. The only option available to the transducer in analyzing EX(6)b is to use transitions based on CO8 to treat it as a sequence of two *NP*s (*the boy, the girl*), followed by a simple *S (the teacher noticed)*, and a *V (knew)*.

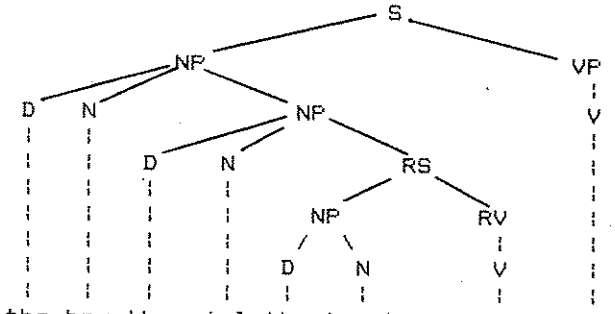
Suppose we allow the transducer to use the transition TS(6)e anyway. The analysis of EX(6)b would then proceed as in FG(6)b2.

FG(6) b. 2. Continuation of the analysis of EX(6)b by FT(6).

STEP	READ-IN	TO-STATE	OUTPUT	RULE-FROM
7.		NP01NP00NP02N11	NP\NP	TS(6)e
8.		NP01NP00NP12	NP\N	TS(2)e2
9.	noticed	NP01NP00RS01V11	RS/NP, V/noticed	TS(6)b
10.		NP01NP00RS01RV11	RV/V	TS(6)a
11.		NP01NP00RS11	RS\RV	TS(6)d2
12.		NP11	NP\RS	TS(6)d1
13.	knew	S01V11	V/knew	TS(2)d2
14.		S01VP11	VP/V	TS(2)c
15.		S11	S\VP	TS(2)e1
16.	#	F		TS(2)b2

The sequence of output statements in FG(6)b1-2 is equivalent to the tree diagram in SD(2)b2.

SD(6) b. 2. Analysis of EX(6)b in FG(6)b1-2.



SD(6)b2 presents an uninterpretable parse for EX(6)b, not only with respect to GR(6), but also with respect to the grammar of English. Nevertheless, there is some reason to believe that native speakers of English occasionally misanalyze expressions with second or higher degrees of center embedding along these lines (Kac 1981).* This suggests either that English grammar contains rules that legitimate transitions like TS(6)e, or that the collapse condition should be relaxed further to legitimate them.

As we mentioned in section 1, the construction can be easily modified to enable the transducer to parse correctly expressions with greater than first degree center embedding. To enable FT(6), for example, to parse expressions of IL(6) like EX(6)b, with second degree center embedding, but fail to parse expressions like

*Note also that FT(6), with the transition TS(6)e, would accept the ungrammatical string EX(6)b' as an *NP*, and associate with it the structural description of the corresponding substring in EX(6)b.

EX(6) b'. the boy the girl the teacher noticed #

Thus not only would FT(6) fail to analyze all the expressions of IL(6), but it also would accept expressions of the complement of IL(6) with respect to UL(6).

EX(6)c, with greater than second degree center embedding, first we increase the maximum value that the *y*-attribute can have from 3 to 4. Second we modify the collapse condition so that it is not invoked until an element of the form *A04* appears in a state. However, we do not modify the collapse chain itself; we continue to connect up only the elements from the incomplete root immediately preceding *A02* to the incomplete element (root or descendant) immediately preceding *A03*. The latter is given a new *y*-attribute value of 2, as before, and the former *A04* is changed to *A03*. In this way the top and the two bottom constituents of the type *A* remain incomplete roots and hence are available for further attachment. As Glenn Blank has pointed out to us, there is considerable evidence that many native speakers of English parse multiply center embedded expressions in this way.

5. FINITE TRANSDUCERS AS A THEORY OF NATURAL LANGUAGE

The class of finite transducers is a theory of grammar which is capable of representing the structures of the acceptable expressions of a natural language as adequately as the class of context-free phrase-structure grammars, and presumably as adequately as other, more powerful, classes.* It differs from these other theories in the inability of its grammars to represent the struc-

*The class of context-free phrase-structure grammars is held to be inadequate as a theory of natural languages because of the inability of those grammars to recognize structures with greater than some fixed finite degree of crossing dependencies (see Postal and Langendoen (1984); Pullum (1984)). However, we may assume that there is a limit on the degree of crossing permitted in acceptable expressions and that finite transducers can be designed that recognize expressions with less than that degree of crossing. Similarly, all generative theories are held to be inadequate as theories of natural languages because of their inability to express the generalization that for any set of constituents of a given type in a natural language, another constituent of that type, consisting of the coordination of the elements of that set, also belongs to the language (Langendoen and Postal, 1984). Again, we may assume that there is some limit on the size of the set in acceptable coordinate structures which allows them to be recognized by a finite transducer.

tures of unacceptable expressions with greater than some fixed, finite degree of center embedding (or with greater than some fixed, finite degree of some other structural property that finite transducers cannot compute in the limit, such as crossing dependencies; see note 9). This limitation has led most linguists to reject the theory of finite transducers as providing an adequate basis for a theory of natural languages. However, if a theory of natural language is intended specifically to account for the tacit linguistic knowledge of human beings (their linguistic competence), there are no grounds for the rejection, because people manifest no tacit knowledge whatever of the grammatical structures of the unacceptable expressions that manifest multiple degrees of center embedding.

The claim that natural languages contain expressions that are unacceptable and that lie outside the bounds of human tacit knowledge can only be maintained if a distinction is drawn between a natural language and a person's tacit knowledge of that language. For example, suppose it is argued that in English, a relative clause in which the direct object has been relativized (a constitutent of the type *RS* in section 4.5 above) can modify any subject noun whatever. Then clearly English contains expressions with every finite degree of center embedding and no grammar of English therefore can be a finite transducer. However, it does not follow that the internalized grammars of native speakers of English cannot be finite transducers, since those individuals have tacit knowledge of only those expressions with up to some fixed, finite degree of center embedding. Their internalized grammars cannot contain a single rule which permits relative clauses to be attached to nouns, but

rather must contain separate rules depending on the grammatical configuration in which those nouns appear (unembedded, in a clause modifying an unembedded noun, in a clause in a clause modifying an unembedded noun, etc.), though these rules may be schematized as in FT(6) in section 4.5.

If the domain of linguistics is the study of the growth and structure of the internalized grammars of human beings, as Chomsky (1980) has proposed, then the theory of finite transducers constitutes an adequate basis for linguistic theory, though of course much work remains to be done to restrict the theory even further, so that the class of possible grammars begins to converge on the class of grammars than can be induced in human beings upon exposure to primary linguistic data.

It seems to us, however, preferable, to maintain the traditional distinction between a language and a person's tacit knowledge of that language. First, as we showed above in several places, it is possible for a person to be misled by his or her internalized grammar about the grammatical nature of particular expressions. However, the only certain way to judge that a person is misled (since all native speakers may be misled in the same way) is by appeal to an external standard such as that provided by the language itself, apart from any one individual's knowledge of it. Second, it is only in the grammar of the language itself, as opposed to the individual internalized grammars of native speakers, that certain generalizations about linguistic structures can be expressed, such as that relative clauses can freely modify nouns in English. One comes to this knowledge not by the passive route of ordinary language acquisition,

but either by explicit instruction or by one's own rational reflection on language. The results of these efforts cannot be dismissed as mere epiphenomena, since once recognized, the generalizations contained in the grammars of natural languages have as much force as the ones that are encoded in one's internalized grammars. Moreover, the distinction between acceptability and unacceptability (now to be construed necessarily as the distinction between grammaticality and ungrammaticality) projects in a perfectly lawful manner from the primary linguistic data, just as before. Thus we conclude that the theory of finite transducers (or a more restricted version of that theory) is appropriate as a theory of a person's knowledge of a natural language, but that a more powerful theory is needed as a theory of natural language itself.

REFERENCES

Brame, Michael. 1979. "A note on COMP S grammar vs. sentence grammar." *Linguistic analysis* 5.383-6.

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1959. "On certain formal properties of grammar." *Information and control* 2.137-67.

Chomsky, Noam. 1980. *Rules and representations*. New York: Columbia University Press.

Gordon, Reena. 1982. "The listener resolves an ambiguity." Unpublished paper for the Westinghouse science talent search.

Kac, Michael. 1981. "Center-embedding revisited." *Proceedings of the third annual conference of the Cognitive Science Society*.123-4.

Kimball, John. 1974. "Seven principles of surface structure parsing in natural language." *Cognition* 2.15-47.

Krauwer, Steven, and Louis des Tombe. 1980. "The finite state transducer as a theory of language." *Utrecht working papers in linguistics* 9.1-86.

Krauwer, Steven, and Louis des Tombe. 1981. "Transducers and grammars as theories of language." *Theoretical linguistics* 8.173-202.

Langendoen, D. Terence. 1961. "Structural descriptions for sentences generated by non-self-embedding constituent structure grammars." Unpublished S.B. thesis, MIT.

Langendoen, D. Terence. 1975. "Finite state parsing of phrase-structure languages and the status of readjustment rules in grammar." *Linguistic inquiry* 6.533-54.

Langendoen, D. Terence. 1979. "On the assignment of constituent structures to the sentences generated by a transformational grammar." *CUNYforum* 7-8.1-32.

Langendoen, D. Terence. 1982. "The grammatical analysis of texts." In *Text processing: Proceedings of Nobel symposium 51*, ed. by Sture Allén. Stockholm: Almqvist and Wiksell.

Langendoen, D. Terence, and Yedidyah Langsam. 1984. "The representation of constituent structures for finite-state parsing." *Proceedings of COLING84*.24-7. Morristown, NJ: Association for Computational Linguistics.

Langendoen, D. Terence, and Paul M. Postal. 1984. *The vastness of natural languages*. Oxford: Basil Blackwell.

Langsam, Yedidyah, Moshe Augenstein and Aaron Tenenbaum. 1985. *Data structures for personal computers*. Englewood Cliffs, NJ: Prentice-Hall.

Postal, Paul M. and D. Terence Langendoen. 1984. "English and the class of context-free languages." *Computational linguistics*.10.177-81.

Pullum, Geoffrey K. 1984. "Syntactic and semantic parsability." *Proceedings of COLING84*.112-22. Morristown, NJ: Association for Computational Linguistics.