WORD-ORDER VARIATION IN NATURAL LANGUAGE GENERATION[1]

Aravind K. Joshi
Department of Computer and Information Science
Room 555 Moore School
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

In natural language generation the grammatical component has to be systematically interfaced to the other components of the system, for example, the planning component. Grammatical formalisms can be studied with respect to their suitability for generation. The tree adjoining grammar (TAG) formalism has been previously studied in terms of incremental generation. In this paper, the TAG formalism has been investigated from the point of view of its ability to handle word-order variation in the context of generation. Word-order cannot be treated as a last minute adjustment of a structure; this position is not satisfactory cognitively or computationally. The grammatical framework has to be able to deal with the word-order phenomena in a way such that it can be systematically interfaced to the other components of the generation system.

1 Introduction

Natural language generation is a very active area in AI research in natural language processing. In principle, comprehension and generation can be viewed as inverses. However, there are some interesting asymmetries. In comprehension, it may be possible, under certain circumstances, to eschew structural (grammatical) information by the use of other knowledge sources. However, in generation, no matter how much higher level knowledge is available, it is not possible to bypass the grammatical component, as the output has to be well-formed and acceptable to the user[2] What this implies is that the grammatical component has to be systematically interfaced to the other components of the generation system, for example, the planning component.

Grammatical formalisms can be viewed as neutral with respect to comprehension or generation, or they may be investigated from the point of view of their suitability for comprehension and generation separately. Although the view that grammatical formalisms can be neutral with respect to generation or comprehension is viable from a purely theoretical perspective, we do not think it is justified cognitively and computationally. This is because comprehension may be largely heuristic but generation is not. Therefore, generation requires a systematic interaction between the grammatical component and the planning component, as we have stated above. A particular aspect of this interface is a kind of flexibility that leads to incremental generation, including the possibility of detaching part of the representation produced by the planner for the generation of a sentence, and using it for the generation of the next sentence, without affecting the well-formedness of the first sentence.

In an earlier paper, Joshi (1986) has investigated the Tree Adjoining Grammar (TAG) formalism from the point of generation. This formalism has been investigated extensively by Joshi and his co-workers (e.g., Joshi, Levy and Takahashi (1975), Joshi (1983, 1985), Kroch and Joshi (1986), Kroch (1986), Vijay-shanker, Weir, and Joshi (1986), and other works). In Joshi (1986), the TAG formalism was studied from the point of its suitability for incremental generation. McDonald and Pustejovsky (1985) have also investigated the TAG formalism for generation with respect to the MUMBLE system of McDonald. In Joshi (1986), the problem of word-order variation in generation was raised and briefly discussed. The main goal of this paper is to investigate, in some detail, the TAG formalism from the point of view of its ability to handle word-order variation in the context of generation. We will also discuss the relationship of our work to other formalisms, particularly with respect to the extent to which they can deal with the issues discussed in this paper. Specifically, we will consider the context-free grammar based formalism such as the Generalized Phrase Structure Grammar (GPSG), and the Functional Unification Grammar (FUG) of Kay, which has been used in some generation systems (e.g., in the TEXT system of McKeown (1984)).

In Section 2, we will give a brief introduction to the TAG formalism together with some examples. In Section 3 we will deal with the problem of word-order variation.

2 A short description of tree adjoining grammars (TAG)

The main characteristics of TAG's are as follows: 1) TAG is a tree generating system. It consists of a finite set of elementary trees (elaborated up to preterminal (terminal) symbols) and a composition operation (adjoining) which builds trees out of elementary trees and trees derived from elementary trees by adjoining. A TAG should be viewed primarily as a tree generating system in contrast to a string generating system such as a context-free grammar or some of its extensions. 2) TAG's factor recursion and dependencies in a novel way. The elementary trees are the domain of dependencies which are statable as co-occurence relations among the elements of the elementary trees and also relations between elementary trees. Recursion enters via the operation of adjoining. Adjoining preserves the dependencies. Localization of dependencies in this manner has both linguistic and computational significance. Such localization cannot be achieved directly in a string generating system. 3) TAG's are more powerful than context-free grammars, but only "mildly" so. This extra power of TAG is a direct corollary of the way TAG factors recursion and dependencies.

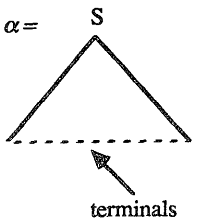
[1]This work was partially supported by ARO grant DAA29-84-9-0027, NSF grants MCS-82-07294, DCR-84-10413, and DARPA grant N00014-85-K-0018. I want to thank Mark Steedman, K. Vijay-Shanker, and D. Weir for their valuable comments.

[2]One might think that the use of templates would avoid this problem; however, this approach is very limited and certainly fails to provide textual coherence.

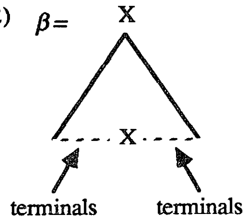
2.1 Tree adjoining grammar formalism

A *tree adjoining grammar (TAG)* G = (I, A) where I and A are finite sets of *elementary trees*. The trees in I will be called the *initial trees* and the trees in A, the *auxiliary trees*. A tree α is an initial tree if it is of the form in (1) and a tree β is an auxiliary tree if it is of the form in 2:

(1) $\alpha =$



(2) $\beta =$



That is, the root node of α is labelled S and the frontier nodes are all terminals, and the root node of β is labelled X where X is a non-terminal and the frontier nodes are all terminals except one which is labelled X, the same label as that of the root. The node labelled X on the frontier will be called the "foot node" of β. The internal nodes are non-terminals. The initial and the auxiliary trees are not constrained in any manner other than as indicated above. The idea, however, is that both the initial and auxiliary trees will be **minimal** in some sense. An initial tree will correspond to a **minimal** sentential tree (i.e., without recursing on any non-terminal) and an auxiliary tree, with root and foot node labelled X, will correspond to a **minimal** recursive structure that must be brought into the derivation, if one recurses on X.

We will now define a composition operation called **adjoining** (or **adjunction**), which composes an auxiliary tree β with a tree γ. Let γ be a tree containing a node (address n) bearing the label X and let β be an auxiliary tree whose root node is also labelled X. (Note that β must have, by definition, a node (and only one such) labelled X on the frontier.) Then the adjunction of β to γ at node n will be the tree γ' that results when the following operation is carried out: 1) The sub-tree of γ at n, call it t, is excised; 2) The auxiliary tree β is attached at n; 3) The sub-tree t is attached to the foot node of β.

Figure 1 illustrates this operation.

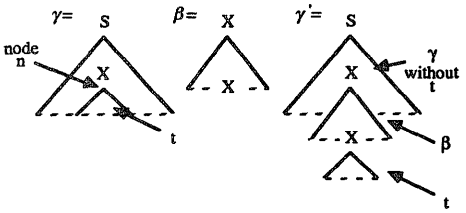


Figure 1

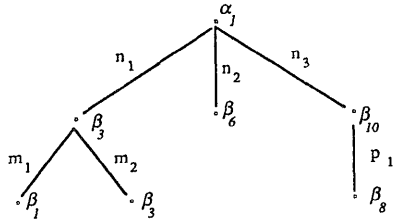
The intuition underlying the adjoining operation is a simple one, but the operation is distinct from other operations on trees that have been discussed in the literature. In particular, we want to emphasize that adjoining is not a substitution operation. **Strictly speaking adjoining is an operation defined between an elementary tree γ and an auxiliary tree (β) which is adjoined to γ at an address n in γ.**

In each elementary tree, any two nodes (or any set of nodes) are <u>dependent</u> simply by virtue of the fact that they belong to the same tree. Of course, some specific dependencies are of interest. These are indicated by co-indexing the nodes (or showing a link between nodes).

2.2 Derivation in a TAG

Although we shall not describe formally the notion of derivation in a TAG, we want to give the reader a more precise understanding of the concept than (s)he might form from description of the operation of adjoining. Adjoining is an operation defined on an elementary tree, say γ, an auxiliary tree, say β, and a node (i.e., an address) in γ, say **n**. Thus, every instance of adjunction is of the form "β is adjoined to γ at **n**," and this adjunction is always and only subject to the local constraints associated with **n**. Although we very often speak of adjoining a tree to a node in a complex structure, we do so only for convenience. Strictly speaking, adjoining is always at a node in an elementary tree; and, therefore, it is more precise to talk about adjoining at an address in an elementary tree. More than one auxiliary tree can be adjoined to an elementary tree as long as each tree is adjoined at a distinct node. After all these auxiliary trees are adjoined to the elementary tree, only nodes in the auxiliary trees are available for further adjunction.

Now suppose that α is an initial tree and that β_1, β_2,... are auxiliary trees in a TAG, G. Then the derivation structure corresponding to the generation of a particular tree and the correspondence string in L(G) might look as follows:



α_1 is an initial tree. β_3, β_6 and β_{10} are adjoined at nodes $\mathbf{n_1}$, $\mathbf{n_2}$, and $\mathbf{n_3}$ respectively in α_1, where $\mathbf{n_1}$, $\mathbf{n_2}$, and $\mathbf{n_3}$ are all distinct nodes. β_1 and β_3 are adjoined to β_3 at nodes $\mathbf{m_1}$ and $\mathbf{m_2}$ respectively. Again, $\mathbf{m_1}$ and $\mathbf{m_2}$ are distinct. β_6 has no further adjunctions but β_8 is adjoined to β_{10} at node $\mathbf{p_1}$. This is a top-down derivation, a bottom-up derivation can be defined also and it is more appropriate for the multicomponent adjunction discussed in Kroch and Joshi (1986). Note that the derivation structure D implicitly characterizes the **surface** tree that is generated by it. D also serves as the basis for defining a compositional semantic interpretation (Vijay-Shanker 1986). In this way the derivation structure can be seen as the basic formal object constructed in the course of sentence generation. Associated with it will be two mappings, one to a surface syntactic tree and the other to a semantic interpretation, as below:

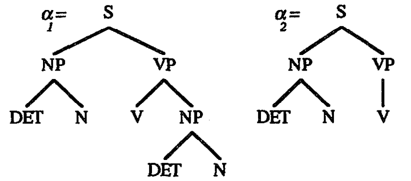
derivation structure

surface tree <--- ---> semantic interpretation

2.3 Some linguistic examples

We will give some simple linguistic examples that illustrate the applicability of the TAG formalism to the description of natural language phenomena. Let G = (I, A) be a TAG where I is the set of initial trees and A is the set of auxiliary trees. We will list only some of the trees in I and A, those relevant to the derivation of our illustrative sentences. Rather than introduce all these trees at once, we shall introduce them as necessary.

I (initial trees) :



Tree α_1 corresponds to a "minimal sentence" with a transitive verb, as in (1); and α_2 corresponds to a minimal sentence with an intransitive verb, as in (2): (1) The man met the woman. and (2) The man fell.

Initial trees as we have defined them require terminal symbols on the frontier. In the linguistic context, the nodes on the frontier will be preterminal lexical category symbols such as N, V, A, P, DET, etc. The lexical items are inserted for each of the preterminal symbols as each elementary tree enters the derivation. Thus, we generate the sentence in (1) by performing lexical insertion on α_1, yielding: (1) The man met the woman.

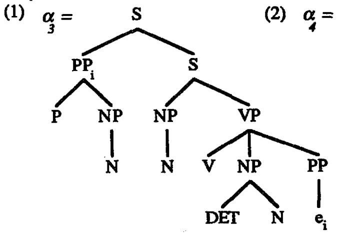
As we continue the derivation by selecting auxiliary trees and adjoining them appropriately, we follow the same convention, i.e., as each elementary tree is chosen, we make the lexical insertions. **Thus in a derivation in a TAG, lexical insertion goes hand in hand with the derivation. This aspect of TAG is highly relevant to generation and is discussed in Joshi (1986).** Each step in the derivation selects an elementary tree together with a set of appropriate lexical items. Note that as we select the lexical items for each elementary tree we can check a variety of constraints, e.g., agreement and subcategorization constraints on the set of lexical items. These constraints can be checked easily because the entire elementary tree that is the domain of the constraints is available as a single unit at each step in the derivation.

As the reader will have noted, we require different initial trees for the sentences "John fell" and "the man fell" because the expansion of NP is different in the two cases. Since the structure of these two sentences is otherwise identical, we cannot be content with a theory that treats the two sentences as unrelated. In a fully articulated theory of grammar employing the TAG formalism, the relationships among initial trees is expressed in an independent module of the grammar that specifies the constraints on possible elementary (initial or auxiliary) trees. And we can even provide schemata or rules for obtaining some elementary structures from others. In any case, these rules are abbreviatory.

The most important point regarding the source of elementary trees is that using the TAG formalism allows us to treat as orthogonal the principles governing the construction of minimal syntactic units and those governing the composition of these units into complex structures.

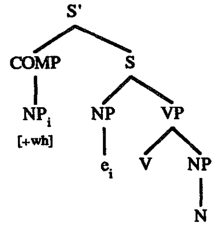
In (1) we give a topicalized structure, and in (2) gives a WH-question:

(1)



to Mary John gave a book

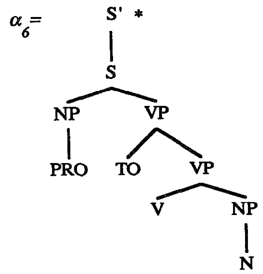
(2)



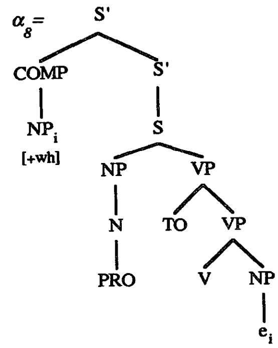
who met Mary

These correspond to (3) To Mary John gave a book, and (4) Who met Mary. Thus far all of the initial trees that we have defined correspond to **minimal** root sentences. We now introduce, in (3) below, some initial trees which are minimal but do not give root sentences. The motivation for introducing these trees will be clear from the examples. Since these trees are not possible root sentences, it is necessary that they undergo at least one adjunction (of a specific type) and the resulting tree becomes a possible independent sentence. This requirement can be very easily stated as a local constraint (discussed in Joshi (1985)). For simplicity, we will neither discuss these constraints nor show them in the diagrams.

(3)



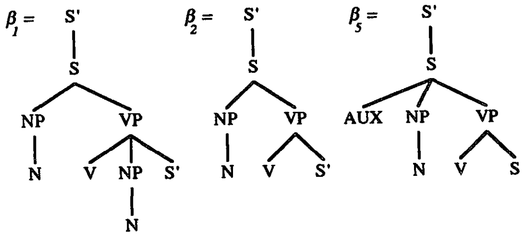
PRO to invite Mary



who PRO to invite

Tree α_6 will be used in the derivation of sentences like (4) and (5): (4) John persuaded Bill PRO to invite Mary. (5) John tried PRO to invite Mary. α_8 will be used in deriving sentences like (6): (6) Who did John try to invite? (PRO stands for the missing subject).

Now we introduce auxiliary trees that will adjoin to the above infinitival initial trees to produce complete independent sentences:

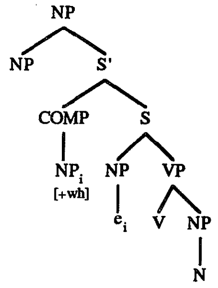


John persuaded Bill S' John tried S' did John try S

The reader can easily check that the sentences (4) - (6) will be derived if the appropriate auxiliary trees in (4) are adjoined at the starred nodes of the initial trees in (3).

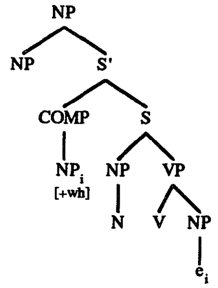
Now let us introduce some auxiliary trees that will allow us to generate sentences with relative clauses:

β_6 =



who met Mary

β_4 =



who met Mary

Tree β_6 can be used to build sentences with subject relatives, as in (6); and β_4 can be used to build sentences with object relatives, as in (7): (6) The boy who met Mary left and (7) The boy who Mary met left.

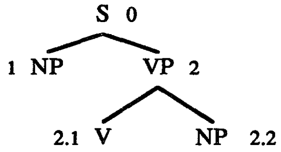
3 Word-order variation

It is well known that all languages allow for word-order variation, but some allow for considerably more than others, the extreme case being the so-called "free" word-order. The linguistic relevance of word-order variation for generation is as follows. First of all, the different word orders (if not all) carry some pragmatic information (topic/new information, for example). The question is at what point the grammatical component should decide on the word order and what point it should reorder the words (or phrases) to reflect this order. The planner can certainly give the pragmatic information to the grammatical component long before all the descriptions are built or even planned. In a TAG, if we work with elementary structures, the grammatical component can use this information immediately and select the appropriate elementary structure. The correct word-order will then be preserved as the sentence is incrementally built. Even if a particular word order has no pragmatic significance, it is difficult to see how the complex patterns can be realized just by reordering the terminals after the sentence is built because many patterns are not realizable by just permuting the siblings of some node. The ability of TAG to specify a given word order at the elementary structure level appears to provide a better interface between the planner and the grammatical component. We will now describe how word-order variation can be handled in a TAG. This feature of TAG is a direct consequence of the extended domain of locality (as compared to CFG) of TAG and the operation of adjoining. FUG shares the first aspect with TAG.

We will now take the elementary trees of a TAG as **elementary domination structures (initial structures and auxiliary structures)** over which linear precedences can be defined. In fact, from now on we will define an **elementary structure** (ES) as consisting of the **domination structure and linear precedences**[3] Thus, α below is the domination structure of an ES.

[3]The idea of factoring constituency (domination) relationships and linear order is basically similar to the ID/LP format of GPSG (Gazdar, Klein, Pullum, and Sag (1985)). However, there are important differences. First the domain of locality is the elementary structures (and not the rewrite rules or local trees), secondly, we have defined the LP for each elementary structure. Of course, a compact description of LP over a set of elementary structures can be easily defined, but when it is compiled out, it will be in the form we have stated here. The ID/LP format of GPSG cannot capture the range of word-order variation permitted by the TAG framework. FUG can capture some word-order variations beyond what ID/LP format can do, but it cannot capture the full range of variations that TAG can.

α =



The addresses for nodes serve to identify the nodes. They are not to be taken as defining the tree ordering. They are just labels for the nodes.

Let LP_1^α be a set of linear precedence statements associated with α

$$LP_1^\alpha = \begin{bmatrix} 1<2 \\ 2.1<2.2 \end{bmatrix}$$

where if $x < y$ (x precedes y) then (1) x and y are nondominating nodes (i.e., x does not dominate y and y does not dominate x) and (2) if x dominates z_1 and y dominates z_2, then $z_1 < z_2$. Note that $<$ is partial.

Note that LP_1^α corresponds exactly to the standard tree ordering. Given LP_1^α the only terminal string that is possible with the ES (α, LP_1^α), where α is the domination structure and LP_1^α is the linear precedence statement.

(1) NP_1 V NP_2

If instead of LP_1^α, we have LP_2^α

$$LP_2^\alpha = \begin{bmatrix} 1<2.1 \\ 2.1<2.2 \end{bmatrix}$$

First note that in $1 < 2.1$, 2.1 is not a sister of 1. We can define precedences between non-sister nodes because the precedences are defined over α, the domain of locality.

Once again, the only terminal string that is possible with the ES (α, LP_2^α) is

(2) NP_1 V NP_2

but there is an important difference between (α, LP_1^α) and (α, LP_2^α), which will become clear when we examine what happens when an auxiliary tree is adjoined to α. Before we discuss this point, let us consider

$$LP_3^\alpha = \phi$$

i.e., there are no precedence constraints. In this case, we will get all six possible orderings (GPSG with the ID/LP format cannot do this).[4]

[4]There are many improvements of the ID/LP framework of GPSG that have been suggested, e.g., Uzkoreit (1986).

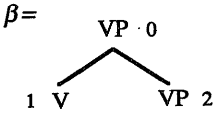
$NP_1\ V\ NP_2$, $NP_1\ NP_2\ V$, $V\ NP_2\ NP_1$, $NP_2\ V\ NP_1$,

$V\ NP_1\ NP_2$, and $NP_2\ NP_1\ V$

Let us return to (α, LP_1^α) and (α, LP_2^α). As we have seen before, both ES give the same terminal string. Now let us consider an ES which is an auxiliary structure (analogous to an auxiliary tree) with an associated LP, LP^β.

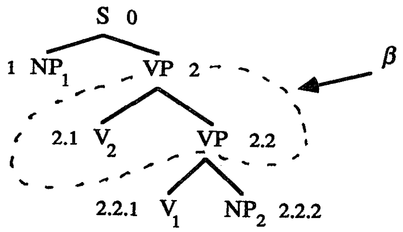
$$LP^\beta = [1 < 2]$$

$\beta=$



When β is adjoined to α at the VP node in α. We have

$\gamma=$



We have put indices on NP and V for easy identification. NP_1, V_1, NP_2 belong to α and V_2 belongs to β. If we have LP_1^α associated with α and LP^β with β, after adjoining the LP's are updated in the obvious manner.

$$LP_1^\alpha = \begin{bmatrix} 1 < 2 \\ 2.2.1 < 2.2.2 \end{bmatrix}$$

$$LP^\beta = [2.1 < 2.2]$$

The resulting LP for γ is

$$LP^\gamma = LP_1^\alpha \cup LP^\beta$$

$$= \begin{bmatrix} 1 < 2 \\ 2.1 < 2.2 \\ 2.2.1 < 2.2.2 \end{bmatrix}$$

Thus γ with LP^γ gives the terminal string

(3) $NP_1\ V_2\ V_1\ NP_2$

Instead of LP_1^α, if we associate LP_2^α with α then after adjoining β to α as before, the updated LP's are

$$LP_2^\alpha = \begin{bmatrix} 1 < 2.2.1 \\ 2.2.1 < 2.2.2 \end{bmatrix}$$

$$LP^\beta = [2.1 < 2.2]$$

The resulting LP for γ is

$$LP^\gamma = LP_2^\alpha \cup LP^\beta$$

$$= \begin{bmatrix} 1 < 2.2.1 \\ 2.1 < 2.2 \\ 2.2.1 < 2.2.2 \end{bmatrix}$$

Thus γ with LP^γ gives the terminal strings

(4) $NP_1\ V_2\ V_1\ NP_2$ (5) $V_2\ NP_1\ V_1\ NP_2$

(4) is the same as (3), but in (5) V_2 has 'moved' past NP_1. If we adjoin β once more to γ at the node VP at 2, then with LP_1^α associated with α, we will get

(6) $NP_1\ V_3\ V_2\ V_1\ NP_2$

and with LP_2^α associated with α, we will get

(7) $NP_1\ V_3\ V_2\ V_1\ NP_2$ (8) $V_2\ NP_1\ V_3\ V_1\ NP_2$

(9) $V_3\ V_2\ NP_1\ V_1\ NP_2$

Let us consider another LP for α, say LP_3^α

$$LP_3^\alpha = [1 < 2.1]$$

Then we have the following terminal strings for α (among others)

(10) $NP_1\ V_1\ NP_2$

(11) $NP_1\ NP_2\ V$

It can be easily seen that given LP_3^αassociated with α and LP^β associated β with $LP^\beta = \varnothing$, after two adjoining with β, we will get (among other strings)

(12) $NP_1\ V_3\ V_2\ V_1\ NP_2$ (13) $NP_1\ V_3\ V_2\ NP_2\ V_1$
(14) $NP_1\ V_3\ NP_2\ V_2\ V_1$ (15) $NP_1\ NP_2\ V_3\ V_2\ V_1$

and, of course, several others. In (13), (14), and (15), NP_2, the complement of V_1 in α has 'moved' past V_1, V_2, and V_3 respectively.

Karttunen (1986) discusses several problems centering around word-order variations in Finnish in the context of a categorial unification grammar. In particular, he deals with auxiliaries and verbs taking infinitival complements. The word order variations lead to dependent elements arbitrarily apart from each other (i.e, long distance dependencies). These long distance dependencies are reminiscent of the long distance dependencies due to topicalization or wh-movement (which we have seen in Section 2.3). There is a difference however. In topicalization or wh-movement, the 'moved' element occupies a grammatically defined position in the structure. The 'moved' element in a long distance dependency of the type Karttunen is concerned about does not move into any structurally defined slot, it 'moves' freely in the host clause.

It can be seen in (7), (8), and (9) that V_2 and V_3 have both 'moved' past NP_1. These 'movements' are not to any grammatically defined positions. Karttunen (1986) gives examples of Finnish auxiliaries which show this long distance behavior and these can be worked out in our framework.

(16) mina en ele aikout ruveta pelaamaan tennista
I not have intend start play tennis
(I have not intended to start to play tennis)

(17) ele mine en aikout reveta pelaamaan tennista

(18) en ele mina aikout reveta pelaamaan tennista.

Further, it can be seen from (13), (14), and (15) that NP_2 (the complement of V_1) can be arbitrarily to the left of V_1 and does not occupy any grammatically defined position. The following examples by Karttunen (1986) can also all be worked into the framework.

(19) mina en ele aikout ruveta pelaamaan tennista
I not have intend start play tennis
(I have not intended to start to play tennis)

(20) en mina ele tennista aikout ruveta pelaamaan

(21) en mina ele aikout tennista ruveta pelaamaan

(22) en mina ele ikout ruveta tennista pelaamaan

Karttunen (1986) uses the devices of type raising (in categorial grammars) or floating types as proposed by Kaplan (unpublished work) to achieve these long distance dependencies.

The elementary structures (ES) with their domination structure and the LP statements factor the constituency (domination) relationships from the linear order. The complex patterns arise due to the nature of the LP and the operation of adjoining. **The main point here is that both the constituency relationships (including the filler-gap relationship) and the linear precedence relationship are defined on the elementary structures. Adjoining preserves these relationships.** We have already seen in Section 2 how the constituency relationships are preserved by adjoining. Now we have seen how the linear precedence relationships are preserved by adjoining. Thus we have a uniform treatment of these two kinds of dependencies; however, the crucial difference between these two kinds (as pointed by Karttunen) clearly shows up in our framework.

The elementary trees of TAG have four properties that can be well matched to incremental building of conceptual structures. These properties are: local definability of all dependencies, locality of feature checking, locality of the argument structure, and preservation of argument structure. All these properties have to do with the constituency structure and "movement" of constituents to grammatically defined positions, as in WH-movement and topicalization. In Joshi (1986) it was shown how these properties help in incremental generation. Our discussion in this section shows that the word-order variation (although distinct from constituent movement as described above) can be localized to elementary trees. The word-orders are specified for the elementary structures and adjoining preserves them. **Thus by working with elementary structures, as described in this section, it appears that we can maintain the incremental generation including word-order variation.**

References

[Appelt, 1985] Appelt, D.E. Planning English Sentences. In *Studies in Natural Language Processing*, Cambridge University Press, Cambridge, 1985.

[Appelt, 1983] Appelt, D.E. Telegram. In *Proceedings IJCAI 1983*. Karlsruhe, August 1983.

[Gazdar *et al*, 1985] Gazdar, G., Klein, J.M.E., Pullum, G.K., and Sag, I.A. *Generalized Phrase Structure Grammar*. Blackwell, Oxford, 1985.

[Joshi, 1985] Joshi, A.K. How much context-sensitivity is required to provide reasonable structural descriptions: tree adjoining grammars. In D. Dowty, L. Karttunen, and A. Zwicky, eds. *Natural Language Processing: Psycholinguistic, Computational And Theoretical Perspectives*. Cambridge University Press, New York, 1985. (Originally presented in May 1983 at the Workshop on Natural Language Parsing at the Ohio State University.)

[Joshi, 1986] Joshi, A.K. The relevance of tree adjoining grammar to generation. In *Proceedings of the 2nd International Workshop on Generation*, Nijmegen, The Netherlands, 1986.

[Karttunen, 1986] Karttunen, L. Radical lexicalism. To appear in M. Baltin and A. Kroch, eds., *New Conceptions of Phrase Structure*, MIT Press, Cambridge, Massachusetts, 1987.

[Kroch and Joshi, 1985] Kroch, A. and Joshi, A.K. Linguistic significance of tree adjoining grammars. To appear in *Linguistics and Philosophy*, 1987.

[Kroch and Joshi, 1986] Kroch, A. and Joshi, A.K. Analyzing extraposition in a tree adjoining grammar. To appear in G. Huck and A. Ojeda, eds., *Syntax and Semantics (Discontinuous Constituents)*, Academic Press, 1986.

[McDonald, 1980] McDonald, D.D. Natural language generation. Ph.D. dissertation, MIT, Cambridge, Mass., 1980.

[McDonald and Pustejovsky, 1985] McDonald, D.D. and Pustejovsky, J. TAG's as a grammatical formalism for generation. In *Proceedings of the 23rd Association for Computational Linguistics (ACL)*, Chicago, June, 1985.

[McKeown, 1985] McKeown, K.R. Text Generation. In *Studies in Natural Language Processing*, Cambridge University Press, 1985.

[Pullum, 1982] Pullum, G.K. Free word order and phrase structure rules. In J. Pustejovsky and P. Sells, eds., *Proceedings of NELS (North Eastern Linguistic Society)*, Amherst, Massachusetts, 1982.

[Shieber, 1986] Shieber, S. *Unification Based Approaches to Grammars*, University of Chicago Press, Chicago, Illinois, 1986.

[Uzkoreit, 1986] Uzkoreit, H. Constraints on order. Technical Report CSLI-86-46, Center for Study of Language and Information (CSLI), Stanford University, 1986.

[Vijay-Shanker and Joshi, 1985] Vijay-Shanker, K. and Joshi, A.K. Some computationally significant properties of tree adjoining grammars. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, June, 1985.