The Relationship Between Tree Adjoining Grammars And Head Grammars†

D.J. Weir K.Vijay-Shanker A.K. Joshi

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract

We examine the relationship between the two grammatical formalisms: Tree Adjoining Grammars and Head Grammars. We briefly investigate the weak equivalence of the two formalisms. We then turn to a discussion comparing the linguistic expressiveness of the two formalisms.

1 Introduction

Recent work [9,3] has revealed a very close formal relationship between the grammatical formalisms of Tree Adjoining Grammars (TAG's) and Head Grammars (HG's). In this paper we examine whether they have the same power of linguistic description. TAG's were first introduced in 1975 by Joshi, Levy and Takahashi[1] and investigated further in [2,4,8]. HG's were first introduced by Pollard[5]. TAG's and HG's were introduced to capture certain structural properties of natural languages. These formalisms were developed independently and are notationally quite different. TAG's deal with a set of elementary trees composed by means of an operation called **adjoining**. HG's maintain the essential character of context-free string rewriting rules, except for the fact that besides concatenation of strings, string **wrapping** operations are permitted. Observations of similarities between properties of the two formalisms led us to study the formal relationship between these two formalisms and the results of this investigation are presented in detail in [9,3]. We will briefly describe the formal relationship established in [9,3], showing TAG's to be equivalent to a variant of HG's. We argue that the relationship between HG's and this variant of HG's called Modified Head Grammars (MHG's) is very close.

Having discussed the question of the *weak* equivalence of TAG's and HG's, we explore, in Sections 4 and 5, what might be loosely described as their *strong* equivalence. Section 4 discusses consequences of the substantial notational differences between the two formalisms. In Section 5, with the use of several examples of analyses (that can not be given by CFG's), we attempt to give cases in which they have the ability to make similar analyses as well as situations in which they differ in their descriptive power.

† This work was partially supported by the NSF grants MCS-82-19116-CER, MCS-82-07294 and DCR-84-10413. We are grateful to Tony Kroch and Carl Pollard, both of whom have made valuable contributions to this work.

1.1 Definitions

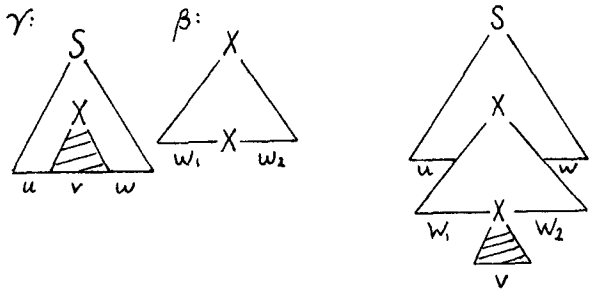
In this section, we shall briefly define the three formalisms: TAG's, HG's, and MHG's.

1.1.1 Tree Adjoining Grammars

Tree Adjoining Grammars differs from string rewriting systems such as Context Free Grammars in that they generate trees. These trees are generated from a finite set of so-called **elementary** trees using the operation of **tree adjunction**. There are two types of elementary trees: **initial** and **auxiliary**. Linguistically, initial trees correspond to phrase structure trees for basic sentential forms, whereas auxiliary trees correspond to modifying structures.

The nodes in the frontier of elementary trees are labelled by terminal symbols except for one node in the frontier of each auxiliary tree, the **foot node**, which is labelled by the same nonterminal symbol as the root. Since initial trees are sentential, their root is always labelled by the nonterminal S.

We now describe the adjoining operation. Suppose we adjoin an auxiliary tree β into a sentential tree γ. The label of the node at which the adjoining operation takes place must be the same as the label of the root (and foot) of β. The subtree under this node is excised from γ, the auxiliary tree β is inserted in its place and the excised subtree replaces the foot of β. Thus the tree obtained after adjoining β is as shown below.



The definition of adjunction allows for more complex constraints to be placed on adjoining. Associated with each node is a **selective adjoining** (SA) constraint specifying that subset of the auxiliary tree which can be adjoined at this node. If the SA constraint specifies an empty subset of trees, then we call this constraint the **Null Adjoining** (NA) constraint. If the SA constraint specifies the entire set of auxiliary tree whose root is labelled with the appropriate nonterminal, then by convention we will not specify the SA constraint. We also allow **obligatory adjoining**(OA) constraints at nodes, to ensure that an adjunction is obligatorily performed at these nodes. When we adjoin an auxiliary tree β in a tree γ those nodes in the resulting tree that do not correspond to nodes of β, retain those constraints appearing in γ. The remaining nodes have the same constraints as those for the corresponding nodes of β.

1.1.2 Head Grammars

Head Grammars are string rewriting systems like CFG's, but differ in that each string has a distinguished symbol corresponding to the head of the string. These are therefore called **headed strings**. The formalism allows not only concatenation of headed strings but also so-called **head wrapping** operations which split a string on one side of the head and place another string between the two substrings. We use one of two notations to denote headed strings: when we wish to explicitly mention the head we use the representation $w_1\overline{a}w_2$; alternatively, we simply denote a headed string by $\overline{w}$. Productions in a HG are of the form $A \rightarrow f(\alpha_1, \ldots, \alpha_n)$ or $A \rightarrow \alpha_1$ where: A is a nonterminal; α_i is either a nonterminal or a headed string; and f is either a concatenation or a head wrapping operation. Roach[6] has shown that there is a normal form for Head Grammars which uses only the following operations.

$$LC1(u_1\overline{a_1}u_2, v_1\overline{a_2}v_2) = u_1\overline{a_1}u_2v_1a_2v_2$$
$$LC2(u_1\overline{a_1}u_2, v_1\overline{a_2}v_2) = u_1a_1u_2v_1\overline{a_2}v_2$$
$$LL1(u_1\overline{a_1}u_2, v_1\overline{a_2}v_2) = u_1\overline{a_1}v_1a_2v_2u_2$$
$$LL2(u_1\overline{a_1}u_2, v_1\overline{a_2}v_2) = u_1a_1v_1\overline{a_2}v_2u_2$$
$$LR1(u_1\overline{a_1}u_2, v_1\overline{a_2}v_2) = u_1v_1a_2v_2\overline{a_1}u_2$$
$$LR2(u_1\overline{a_1}u_2, v_1\overline{a_2}v_2) = u_1v_1\overline{a_2}v_2a_1u_2$$

1.1.3 Modified Head Grammars

Pollard's definition of headed strings includes the headed empty string ($\overline{\lambda}$). However the term $fi(\overline{w_1}, \ldots, \overline{w_i}, \ldots, \overline{w_n})$ is undefined when $\overline{w_i} = \overline{\lambda}$. This nonuniformity has led to difficulties in proving certain formal properties of HG's[6]. MHG's were considered to overcome these problems. Later in this paper we shall argue that MHG's are not only close to HG's formally, but also that they can be given a linguistic interpretation which retains the essential characteristics of HG's. It is worth noting that the definition of MHG's given here coincides with the definition of HG's given in [7].

Instead of headed strings, MHG's use so-called **split strings**. Unlike a headed string which has a distinguished symbol, a split string has a distinguished *position* about which it may be split. In MHG's, there are 3 operations on split strings: W, $C1$, and $C2$. The operations $C1$ and $C2$ correspond to the operations $LC1$ and $LC2$ in HG's. They are defined as follows:

$$C1(w_{1\uparrow}w_2, u_{1\uparrow}u_2) = w_{1\uparrow}w_2u_1u_2$$

$$C2(w_{1\uparrow}w_2, u_{1\uparrow}u_2) = w_1w_2u_{1\uparrow}u_2$$

Since the split point is not a symbol (which can be split either to its left or right) but a position between strings, separate left and right wrapping operations are not needed. The wrapping operation, W, in MHG is defined as follows:

$$W(w_{1\uparrow}w_2, u_{1\uparrow}u_2) = w_1u_{1\uparrow}u_2w_2$$

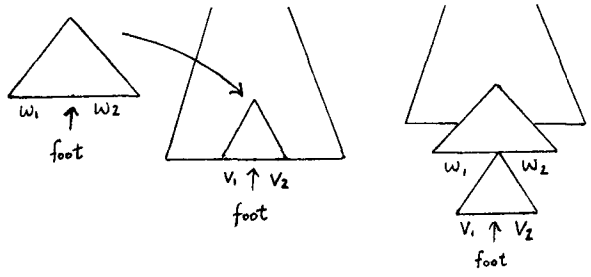
We could have defined two operations $W1$ and $W2$ as in HG. But since $W1$ can very easily be simulated with other operations, we require only $W2$, renamed simply W.

2 MHG's and TAG's

In this section, we discuss the weak equivalence of TAG's and MHG's. We will first consider the relationship between the wrapping operation W of MHG's and the adjoining operation of TAG's.

2.1 Wrapping and Adjoining

The weak equivalence of MHG's and TAG's is a consequence of the similarities between the operations of wrapping and adjoining. It is the roles played by the split point and the foot node that underlies this relationship. When a tree is used for adjunction, its foot node determines where the excised subtree is reinserted. The strings in the frontier to the left and right of the foot node appear on the left and right of the frontier of the excised subtree. As shown in the figure below, the foot node can be thought of as a position in the frontier of a tree, determining how the string in the frontier is split.



Adjoining in this case, corresponds to wrapping $w_{11}w_2$ around the split string $v_{11}v_2$. Thus, the split point and the foot node perform the same role. The proofs showing the equivalence of TAG's and MHG's is based on this correspondence.

2.2 Inclusion of TAL in MHL

We shall now briefly present a scheme for transforming a given TAG to an equivalent MHG. We associate with each auxiliary tree a set of productions such that each tree generated from this elementary tree with frontier w_1Xw_2 has an associated derivation in the MHG, using these productions, of the split string $w_{11}w_2$. The use of this tree for adjunction at some node labelled X can be mimicked with a single additonal production which uses the wrapping operation.

For each elementary tree we return a sequence of productions capturing the structure of the tree in the following way. We use nonterminals that are named by the nodes of elementary trees rather than the labels of the nodes. For each node η in an elementary tree, we have two nonterminal X_η and Y_η: X_η derives the strings appearing on the frontier of trees derived from the subtree rooted at η; Y_η derives the concatenation of the strings derived under each daughter of η. If η has daughters $\eta_1, \ldots, \eta_k$ then we have the production:

$$Y_\eta \rightarrow Ci(X_{\eta_1}, \ldots, X_{\eta_k})$$

where the node η_i dominates the foot node (by convention, we let $i = 1$ if η does not dominate the foot node). Adjunction at η, is simulated by use of the following production:

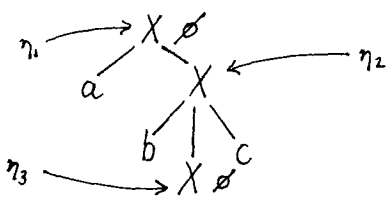
$$X_\eta \rightarrow W(X_\mu, Y_\eta)$$

where μ is the root of some auxiliary tree which can be adjoined at η. If adjunction is optional at η then we include the production:

$$X_\eta \rightarrow Y_\eta.$$

Notice that when η has an NA or OA constraint we omit the second or third of the above productions, respectively.

Rather than present the full details (which can be found in [9,3]) we illustrate the construction with an example showing a single auxiliary tree and the corresponding MHG productions.

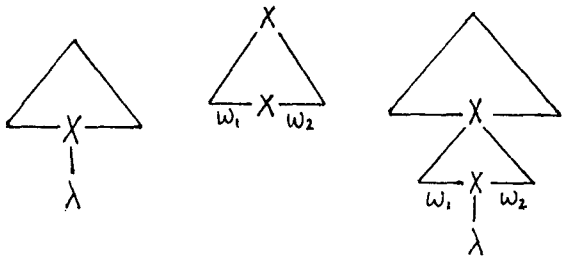


$$\begin{aligned}
&X_{\eta_1} \rightarrow Y_{\eta_1},\\
&Y_{\eta_1} \rightarrow C2(a, X_{\eta_2}),\\
&X_{\eta_2} \rightarrow W(X_{\mu_1}, Y_{\eta_2}),\\
&\qquad\vdots\\
&X_{\eta_2} \rightarrow W(X_{\mu_n}, Y_{\eta_2}),\\
&X_{\eta_2} \rightarrow Y_{\eta_2},\\
&Y_{\eta_2} \rightarrow C2(b, X_{\eta_3}c)\\
&X_{\eta_3} \rightarrow Y_{\eta_3}\\
&Y_{\eta_3} \rightarrow \lambda
\end{aligned}$$

where $\mu_1, \ldots, \mu_n$ are the roots of the auxiliary trees adjoinable at η_2.

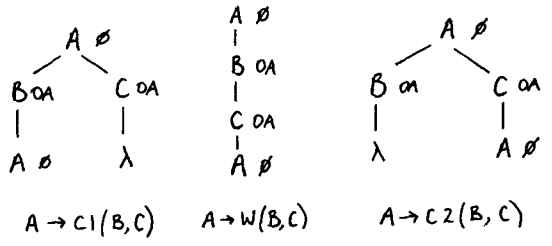
2.3 Inclusion of MHL in TAL

In this construction we use elementary trees to directly simulate the use of productions in MHG to rewrite nonterminals. Generation of a derivation tree in string-rewriting systems involves the substitution of nonterminal nodes, appearing in the frontier of the unfinished derivation tree, by trees corresponding to productions for that nonterminal. From the point of view of the string languages obtained, tree adjunction can be used to simulate substitution, as illustrated in the following example.



Notice that although the node where adjoining occurs does not appear in the frontier of the tree, the presence of the node labelled by the empty string does not effect the string language.

For each production in the MHG we have an auxiliary tree. A production in an MHG can use one of the three operations: $C1$, $C2$, and W. Correspondingly we have three types of trees, shown below.



Drawing the analogy with string-rewriting systems: NA constraints at each root have the effect of ensuring that a nonterminal is rewritten only once; NA constraints at the foot node ensures that, like the nodes labelled by λ, they do not contribute to the strings derived; OA constraints are used to ensure that every nonterminal introduced is rewritten at least once.

The two trees mimicking the concatenation operations differ only in the position of their foot node. This node is positioned in order to satisfy the following requirement: for every derivation in the MHG there must be a derived tree in the TAG for the same string, in which the foot is positioned at the split point.

The tree associated with the wrapping operation is quite different. The foot node appears below the two nodes to be expanded because the wrapping operation of MHG's corresponds to the $LL2$ operation of HG's in which the head (split point) of the second argument becomes the new head (split point). Placement of the nonterminal, which is to be wrapped, above the other nonterminal achieves the desired effect as described earlier.

While straightforward, this construction does not capture the linguistic motivation underlying TAG's. The auxiliary trees directly reflect the use of the concatenation and the wrapping operations. As we discuss in more detail in Section 4, elementary trees for natural languages TAG's are constrained to capture meaningful linguistic structures. In the TAG's generated in the above construction, the elementary trees are incomplete in this respect: as reflected by the extensive use of the OA constraints. Since HG's and MHG's do not explicitly give minimal linguistic structures, it is not surprising that such a direct mapping from MHG's to TAG's does not recover this information.

3 HG's and MHG's

In this section, we will discuss the relationship between HG's and MHG's. First, we outline a construction showing that HL's are included in MHL's. Problems arise in showing the inclusion in the other direction because of the nonuniform way in which HG's treat the empty headed string. In the final part of this section, we argue that MHG's can be given a meaningful linguistic interpretation, and may be considered essentially the same as HG's.

3.1 HL's and MHL's

The inclusion of HL's in MHL's can be shown by constructing for every HG, G, an equivalent MHG, G'. We now present a short description of how this construction proceeds.

Suppose a nonterminal X derives the headed string $w_1\overline{h}w_2$. Depending on whether the left or right wrapping operation is used, this headed string can be split on either side of the head. In fact, a headed string can be split first to the right of its head and then the resulting string can be split to the left of the same head. Since in MHG's we can only split a string in one place, we introduce nonterminals $X^{\uparrow h}$, that derive split strings of the form $w_1{\uparrow}w_2$ whenever X derives $w_1\overline{h}w_2$ in the HG. The missing head can be reintroduced with the following productions:

$$X^l \rightarrow W(X^{\uparrow h}, h{\uparrow}) \qquad \text{and} \qquad X^r \rightarrow W(X^{\uparrow h}, {\uparrow}h)$$

Thus, the two nonterminals, X^l and X^r derive $w_1h{\uparrow}w_2$ and $w_1{\uparrow}hw_2$ respectively. Complete details of this proof are given in [3].

We are unable to give a general proof showing the inclusion of MHL's in HL's. Although Pollard[5] allows the use of the empty headed string, mathematically, it does not have the same status as other headed strings. For example, $LC1(\overline{\lambda}, \overline{w})$ is undefined. Although we have not found any way of getting around this in a systematic manner, we feel that the problem of the empty headed string in the HG formalism does not result from an important difference between the formalisms.

For any particular natural language, Head Grammars for that language appear to use either only the left wrapping operations LLi, or only the right wrapping operations LRi. Based on this observation, we suggest that for any HG for a natural language, there will be a corresponding MHG which can be given a linguistic interpretation. Since headed strings will always be split on the same side of the head, we can think of the split point in a split string as determining the head position. For example, split strings generated by a MHG for a natural language that uses only the left wrapping operations have their split points immediately to the right of the actual head. Thus a split point in a phrase not only defines where the phrase can be split, but also the head of the string.

4 Notational Differences between TAG's and HG's

TAG's and HG's are notationally very different, and this has a number of consequences that influence the way in which the formalisms can be used to express various aspects of language structure. The principal differences derive from the fact that TAG's are a tree-rewriting system unlike HG's which manipulate strings.

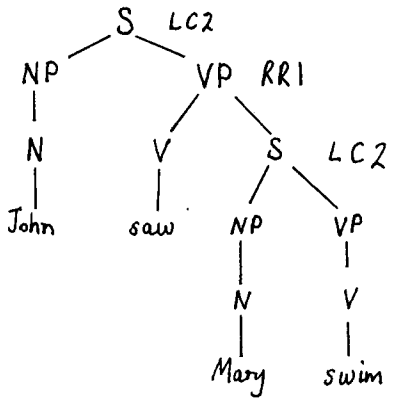
The elementary trees in a TAG, in order to be linguistically meaningful, must conform to certain constraints that are not explicitly specified in the definition of the formal-

ism. In particular, each elementary tree must constitute a *minimal linguistic structure.* Initial trees have essentially the structure of simple sentences; auxiliary trees correspond to minimal recursive constructions and generally constitute structures that act as modifiers of the category appearing at their root and foot nodes.

A hypothesis that underlies the linguistic intuitions of TAG's is that all dependencies are captured within elementary trees. This is based on the assumption that elementary trees are the appropriate domain upon which to define dependencies, rather than, for example, productions in a Context-free Grammar. Since in string-rewriting systems, dependent lexical items can not always appear in the same production, the formalism does not prevent the possibility that it may be necessary to perform an unbounded amount of computation in order to check that two dependent lexical items agree in certain features. However, since in TAG's dependencies are captured by bounded structures, we expect that the complexity of this computation does not depend on the derivation. Features such as agreement may be checked within the elementary trees (instantiated up to lexical items) without need to percolate information up the derivation tree in an unbounded way. Some checking is necessary between an elementary tree and an auxiliary tree adjoined to it at some node, but this checking is still local and unbounded. Similarly, elementary trees, being minimal linguistic structures, should capture all of the sub-categorization information, simplifying the processing required during parsing. Further work (especially empirical) is necessary to confirm the above hypothesis before we can conclude that elementary trees can in fact capture all the necessary information or whether we must draw upon more complex machinery. These issues will be discussed in detail in a later paper.

Another important feature of TAG's that differentiates them from HG's is that TAG's generate phrase-structure trees. As a result, the elementary trees must conform to certain constraints such as left-to-right ordering and linguistically meaningful dominance relations. Unlike other string-rewriting systems that use only the operation of concatenation, HG's do not associate a phrase-structure tree with a derivation: wrapping, unlike concatenation, does not preserve the word order of its arguments. In the Section 5, we will present an example illustrating the importance of this difference between the two formalisms.

It is still possible to associate a phrase-structure with a derivation in HG's that indicates the constituents and we use this structure when comparing the analyses made by the two systems. These trees are not really phrase-structure trees but rather trees with annotations which indicate how the constituents will be wrapped (or concatenated). It is thus a derivation structure, recording the history of the derivation. With an example we now illustrate how a constituent analysis is produced by a derivation in a HG.



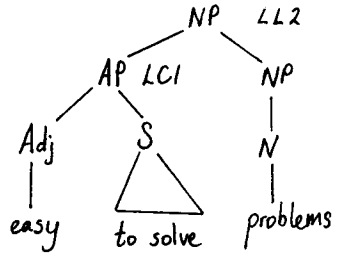
5 Towards "Strong" equivalence

In Section 2 we considered the weak equivalence of the two formalisms. In this section, we will consider three examples in order to compare the linguistic analyses that can be given by the two formalisms. We begin with an example (Example 1) which illustrates that the construction given in Section 2 for converting a TAG into an MHG gives similar structures. We then consider an example (Example 2) which demonstrates that the construction does not always preserve the structure. However, there is an alternate way of viewing the relationship between wrapping and adjoining, which, for the same example, does preserve the structure.

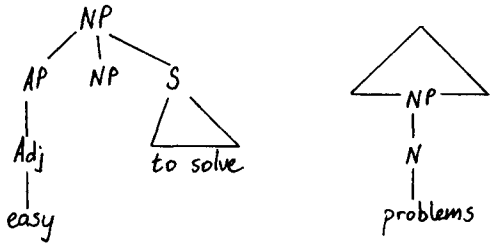
Although the usual notion of strong equivalence (i.e., equivalence under identity of structural descriptions) can not be used in comparing TAG and HG (as we have already indicated in Section 4), we will describe informally what the notion of "strong" equivalence should be in this case. We then illustrate by means of an example (Example 3), how the two systems differ in this respect.

5.1 Example 1

Pollard[5] has suggested that HG can be used to provide an appropriate analysis for *easy problems to solve.* He does not provide a detailed analysis but it is roughly as follows.



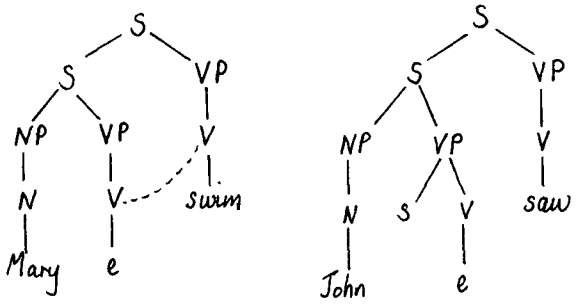
This analysis can not be provided by CFG's since in deriving *easy to solve* we can not obtain *easy to solve* and *problems* as intermediate phrases. The appropriate elementary tree for a TAG giving the same analysis would be:



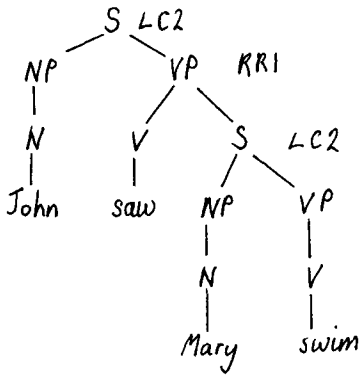
Note that the phrase *easy to solve* wraps around *problems* by splitting about the head and the foot node in both the grammars. Since the conversion of this TAG would result in the HG given above, this example shows that the construction captures the correct correspondence between the two formalisms.

5.2 Example 2

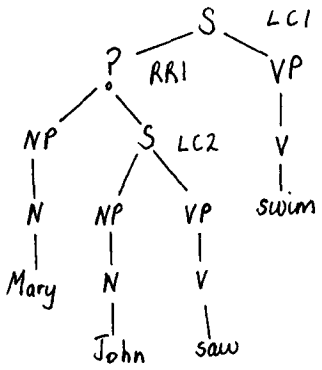
We now present an example demonstrating that the construction does not always preserve the details of the linguistic analysis. This example concerns cross-serial dependencies, for example, dependencies between NP's and V's in subordinate clauses in Dutch (cited frequently as an example of a non-context-free construction). For example, the Dutch equivalent of *John saw Mary swim* is *John Mary saw swim*. Although these dependencies can involve an arbitrary number of verbs, for our purposes it is sufficient to consider this simple case. The elementary trees used in a TAG, G_{TAG}, generating this sentence are given below.



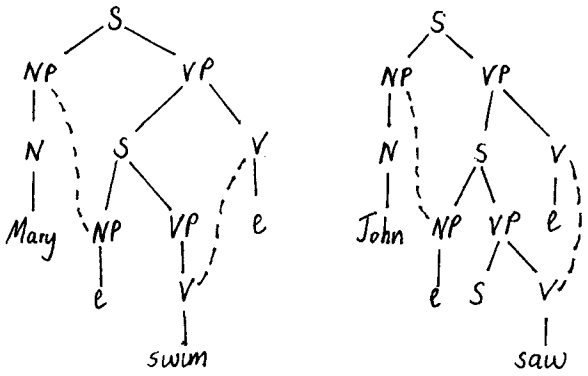
The HG given in [5] (G_{HG}) assigns the following derivation structure (an annotated phrase-structure recording the history of the derivation) for this sentence.



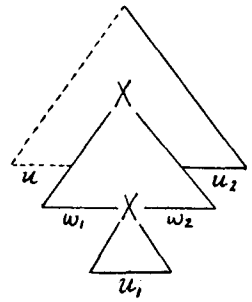
If we use the construction in Section 2 on the elementary trees for the TAG shown above, we would generate an HG, G'_{HG}, that produces the following analysis of this sentence.



This does not give the same analysis as G_{HG}: both G_{HG} and G_{TAG} give intermediate structures in which the predicate *help(Mary swim)* is formed. This then combines with the noun phrase *John* giving the resulting sentence. In the HG G'_{HG} *John* is first combined with *Mary swim*: this is not an acceptable linguistic structure. G'_{HG} corresponds in this sense to the following unacceptable TAG, G'_{TAG}.



Not only does the construction map the acceptable TAG to the unacceptable HG; but it can also be shown that the unacceptable TAG is converted into the acceptable HG. This suggests that our construction does not always preserve linguistic analyses. This arises because the use of wrapping operation does not correspond to the way in which the foot node splits the auxiliary tree in this case. However, there is an alternate way of viewing the manner in which wrapping and adjoining can be related. Consider the following tree.



Instead of wrapping w_1w_2 around u_1 and then concatenating u_2; while deriving the string $w_1u_1w_2u_2$ we could derive the string by wrapping u_1u_2 around w_2 and then concatenating w_1. This can not be done in the general case (for example, when the string u is nonempty).

The two grammars G_{HG} and G_{TAG} can be related in this manner since G_{TAG} satisfies the required conditions. This approach may be interpreted as combining the phrase u_1u_2 with w_2 to form the phrase $u_1w_2u_2$. Relating the above tree to Example 2, u_1 and u_2 correspond to *Mary* and *swim* respectively and w_2 corresponds to *saw*. Thus, *Mary swim* wraps around *saw* to produce the verb phrase *Mary saw swim* as in the TAG G_{TAG} and the HG G_{HG}.

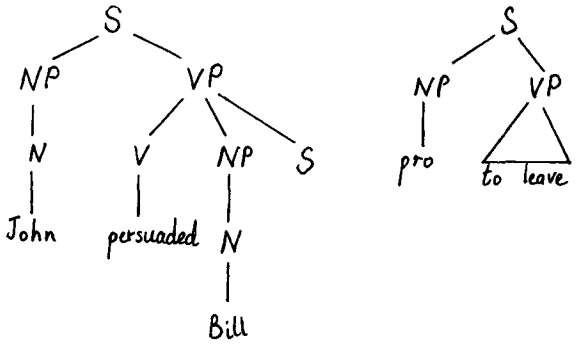
As the previous two examples illustrate, there are two ways of drawing a correspondence between wrapping and adjoining,both of which can be applicable. However, only one of them is general enough to cover all situations, and is the one used in Sections 2 and 3 in discussing the weak equivalence.

5.3 Example 3

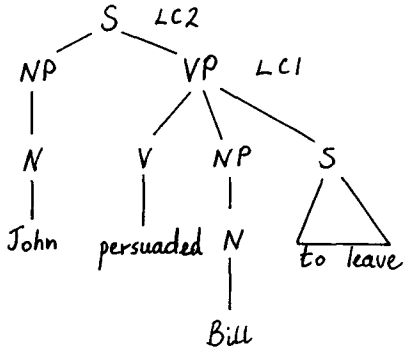
The normal notion of strong equivalence can not be used to discuss the relationship between the two formalisms, since HG's do not generate the standard phrase structure trees (from the derivation structure). However, it is possible to relate the analyses given by the two systems. This can be done in terms of the intermediate constituent structures.

So far, in Examples 1 and 2 considered above we showed that the same analyses can be given in both the formalisms. We now present an example suggesting that this is not always the case. There are certain constraints placed on elementary trees: that they use meaningful elementary trees corresponding to minimal linguistic structures (for example, the verb and all its complements, including the subject complement are in the same elementary tree); and that the final tree must be a phrase-structure tree. As a result, TAG's can not give certain analyses which the HG's can provide, as evidenced in the following example.

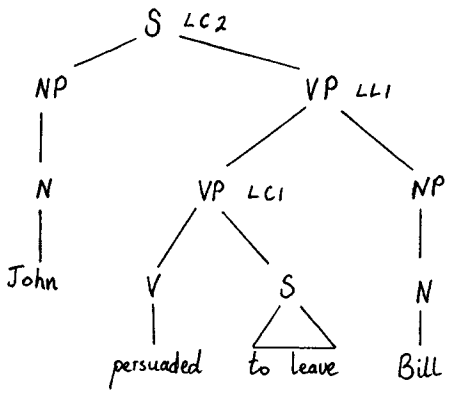
The example we use concerns analyses of *John persuaded Bill to leave.* We will discuss two analyses both of which have been proposed in the literature and have been independently justified. First, we present an analysis that can be expressed in both formalisms. The TAG has the following two elementary trees.



The derivation structure corresponding to this analysis that HG's can give is as follows.



However, Pollard[5] gives another analysis which has the following derivation structure.



In this analysis the predicate *persuade to leave* is formed as an intermediate phrase. Wrapping is then used to derive the phrase *persuade Bill to leave.* To provide such an analysis with TAG's, the phrase *persuade to leave* must appear in the same elementary tree. *Bill* must either appear in an another elementary tree or must be above the phrase *persuade to leave* if it appears in the same elementary tree (so that the phrase *persuade to leave* is formed first). It can not appear above the phrase *persuade to leave* since then the word order will not be correct. Alternatively, it can not appear in a separate elementary tree since no matter which correspondence we make between wrapping and adjoining, we can not get a TAG which has meaningful elementary trees providing the same analysis. Thus the only appropriate TAG for this example is as shown above.

The significance of this constraint that TAG's appear to have (illustrated by Example 3) can not be assessed until a wider range of examples are evaluated from this point of view.

6 Conclusion

This paper focusses on the linguistic aspects of the relationship between Head Grammars and Tree Adjoining Grammars. With the use of examples, we not only illustrate cases where the two formalisms make similar analyses, but also discuss differences in their descriptive power. Further empirical study is required before we can determine the significance of these differences. We have also briefly studied the consequences of the notational differences between the formalisms. A more detailed analysis of the linguistic and computational aspects of these differences is currently being pursued.

References

[1] Joshi, A. K., Levy, L. S., and Takahashi, M. Tree Adjunct Grammars. *Journal of Computer and System Sciences* 10(1), March, 1975.

[2] Joshi, A. K. How Much Context-Sensitivity is Necessary for Characterizing Structural descriptions - Tree Adjoining Grammars. In D. Dowty, L. Karttunen and Zwicky, A. (editors), *Natural Language Processing - Theoretical, Computational and Psychological Perspective.* Cambridge University Press, New York, 1985. originally presented in 1983.

[3] Joshi, A. K., Vijay-Shanker, K., and Weir, D.J. *Tree Adjoining Grammars and Head Grammars.* Technical Report MS-CIS-86-1, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, January, 1986.

[4] Kroch, A. and Joshi, A. K. *Linguistic Relevance of Tree Adjoining Grammars.* Technical Report MS-CIS-85-18, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, April, 1985. also to appear in *Linguistics and Philosophy*, 1986.

[5] Pollard, C. *Generalized Phrase Structure Grammars, Head Grammars and Natural Language.* PhD thesis, Stanford University, August, 1984.

[6] Roach, K. Formal Properties of Head Grammars. 1985. Presented at *Mathematics of Language* workshop at the University of Michigan, Ann Arbor.

[7] Rounds, W. C. LFP: A Logic for Linguistic Descriptions and an Analysis of its Complexity. September, 1985. University of Michigan.

[8] Vijay-Shanker, K. and Joshi, A. K. Some Computational Properties of Tree Adjoining Grammars. In 23^{rd} *meeting of Assoc. of Computational Linguistics*, pages 82–93. July, 1985.

[9] Vijay-Shanker, K., Weir, D. J., and Joshi, A. K. Tree Adjoining and Head Wrapping. In 11^{th} *International Conference on Computational Linguistics.* August, 1986.