Tree Adjoining and Head Wrapping[†]

K. Vijay-Shanker   David J. Weir   Aravind K. Joshi

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

Abstract

In this paper we discuss the formal relationship between the classes of languages generated by Tree Adjoining Grammars and Head Grammars. In particular, we show that Head Languages are included in Tree Adjoining Languages and that Tree Adjoining Grammars are equivalent to a modification of Head Grammars called Modified Head Grammars. The inclusion of MHL in HL, and thus the equivalence of HG's and TAG's, in the most general case remains to be established.

1. Introduction

This paper discusses the relationship between Tree Adjoining Grammars (TAG's) and Head Grammars (HG's). TAG's and HG's, introduced to capture certain structural properties of natural languages, were developed independently. TAG's deal with a set of elementary trees which are composed by means of an operation called **adjoining**. HG's are like Context-free Grammars, except for the fact that besides concatenation of strings, **string wrapping** operations are permitted. TAG's were first introduced in 1975 by Joshi, Levy and Takahashi [3]. Joshi [2] investigated some formal and linguistic properties of TAG's with local constraints. The formulation of local constraints was then modified and formal properties were investigated by Vijay-Shanker and Joshi [9]. The linguistic properties were studied in detail by Kroch and Joshi [5]. HG's were first introduced by Pollard [6] in 1983 and their formal properties were investigated by Roach [7]. It was observed that the two systems seemed to possess similar generative power and since they also appear to have the same closure properties [7,9] as well as similar parsing algorithms [6,9] a significant amount of indirect evidence existed to suggest that they were formally equivalent. In the present paper, we will attempt to provide a characterization of the formal relationship between HG's and TAG's. In [10] we consider various linguistic aspects of the relationship: in particular what might be referred to as the strong equivalence of the two formalisms.

Vijay-Shanker and Joshi [9] provided a brief description of the intuition behind the inclusion of Tree Adjoining Langages (TAL) in the class of languages generated by a variant of HG's called Modified Head Grammars (MHG's). In the present paper, we give a proof of this result as well as a proof for the inclusion of Modified Head Languages (MHL) in TAL: hence we show that MHG's and TAG's are equivalent. This result is presented in section 3. In section 2, we discuss the relationship between MHG's and HG's.

† This work was partially supported by NSF grants MCS-82-19116-CER, MCS-82-07294 and DCR-84-10413. We are grateful to Tony Kroch and Carl Pollard for valuable discussions.

1.1. Tree Adjoining Grammars

Definition 1. *A TAG is a 5-tuple $G = (V_N, V_T, S, I, A)$ where V_N and V_T are finite sets of nonterminals and terminals respectively, S is a distinguished nonterminal, I is a finite set of initial trees and A is a finite set of auxiliary trees.*

Initial trees and auxiliary trees have the following form:

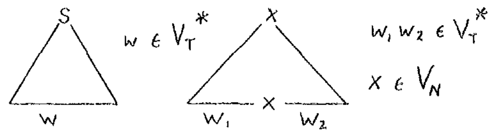


figure 1.1

Except for one node, which is called the **foot node**, all the nodes in the frontier of an auxiliary tree are labelled by terminal symbols. The foot node is labelled by the same nonterminal symbol as the root. All initial and auxiliary trees, referred to as elementary trees, have a height of at least one.

We now define the **adjoining** operation. Let η be some node labelled X in a tree γ. Let β be an auxiliary tree with root and foot labelled by X. The tree obtained by adjoining β at η is given in figure 1.2.

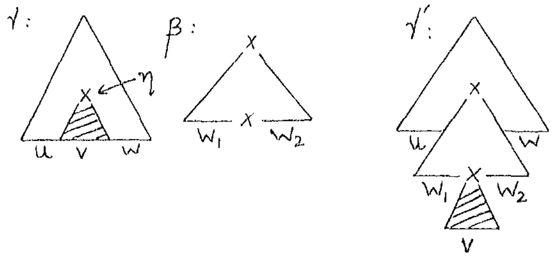


figure 1.2

The subtree under η is excised from γ, the auxiliary tree β is inserted in its place and the excised subtree is inserted at the foot of β.

As defined above it is possible to adjoin any auxiliary tree at a node as long as the label of the node was the same as that of the root and the foot of the auxiliary tree. However, in general, adjoining will be constrained as follows. Associated with each node is a **selective adjoining** (SA) constraint specifying that subset of the auxiliary tree which can be adjoined at this node. Trees can only be included in the SA constraint associated with a particular node if their root and foot are labelled with the same nonterminal that labels the node .There are two special cases: (a) the subset specified by SA is the entire set of auxiliary trees, in this case the entire set need not be explicitly listed; (b) the subset specified is empty i.e., no adjoining is possible. We call this the **null adjoining** (NA) constraint. A node may be associated with a a so-called **obligatory adjoining** (OA) constraint which can be used to ensure that an adjunction is obligatorily performed at a node.

Example 1.1 We now present an example TAG G, which illustrates the notation used to specify the constraints associated with a node. There is one initial tree α and two auxiliary trees β_1 and β_2:

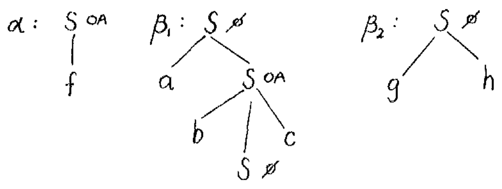


figure 1.3

Having introduced SA and OA constraints, we must extend the definition of the adjunction operation. Suppose we adjoin an auxiliary tree β at a node η of a tree γ producing the tree γ'. For those nodes in γ' that do not correspond to nodes of β, the constraints remain the same as those in γ. The remaining nodes in γ' have the same constraints as those for the corresponding nodes of β. For example, consider a sample derivations in the grammar G as given below in figure 1.4. We use an $*$ to indicate the node at which adjunction is performed.

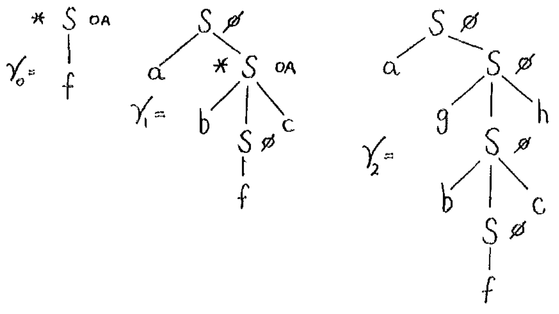


figure 1.4

We will now present an alternative (yet equivalent) definition of the adjoining operation. So far, our definition allowed us to adjoin only with auxiliary trees, and allowed adjunction only into sentential trees. This can be generalized to allow adjunctions of any tree derived from an auxiliary tree into any derived tree. Consider the derivation given in figure 1.4. Given this generalization of adjunction, we can also derive the same tree γ_2 by composing trees in the following sequence.

The derived auxiliary tree γ_1' can be obtained by adjoining β_2 in β_1. γ_1' can then be adjoined in α to give γ_2'. Notice that trees derived from an auxiliary tree β, will always have the property that their root and foot are labelled with the same nonterminal as those of β. Viewing a derivation in this manner considerably simplifies several proofs of formal properties of TAG's [9].

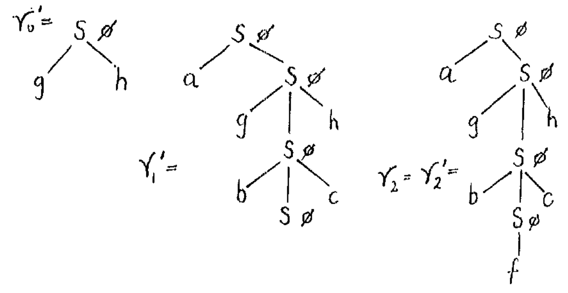


figure 1.5

We use the notation $\mathcal{D}(\gamma)$ to denote the set of trees derived from the elementary tree γ using 0 or more adjunctions.

The **tree set** *$T(G)$ of a TAG G is $T(G) = \bigcup_{\alpha \in I} \mathcal{D}(\alpha)$.*

The **string language** *$L(G)$ generated by a TAG G is given by $L(G) = \{\, w \mid w$ is the frontier of some γ in $T(G)\,\}$*

Now we can see that the language L_1, generated by the example grammar G, is $L_1 = \{\, a^n g b^n f c^n h \mid n \geq 0 \,\}$

It is useful to further generalize the notion of a derived tree to include trees derived from *subtrees* of elementary trees. If η is a node in some elementary tree, then $\mathcal{D}(\eta)$ represents the set of trees derived from the subtree rooted at η. Nodes are represented using an extension of the tree addressing scheme of Gorn [1]. Each node in an elementary tree is given a unique name in the following manner: the pair $\langle \gamma, \epsilon \rangle$ denotes the root of γ; if $\langle \gamma, i \rangle$ is a node in γ, then $\langle \gamma, i \cdot j \rangle$ represents the j^{th} daughter of this node.

1.2. Head Grammars

Before giving the formal definition of Head Grammars, the notion of a **headed string** will be described. A headed string is a string of symbols containing one distinguished symbol referred to as the head of the string. Formally, this can be represented as a pair consisting of a string w and an integer that indicates the position of the head in the string. In this paper, we use one of two notations to denote this string: when we wish to explicitly mention the head we use the representation $w_1 \overline{a} w_2$ where $w_1 a w_2 = w$; alternatively, we can simply denote the headed string by $\overline{w}$. This allows us to denote the headed empty string as $\overline{\lambda}$.

Definition 2. *A Head Grammar, G, is given by a 4-tuple (V_N, V_T, S, P). Productions in P are of the form: $A \rightarrow f(\alpha_1, \ldots, \alpha_n)$ or $A \rightarrow \alpha_1$ where $A \in V_N$, α_i either belongs to V_N or is a headed string.*

$$f \in \bigcup_{i \geq 1} \{\, LCi, LLi, LRi, RCi, RLi, RRi \,\}$$

We now define the operations LCi, LLi, and LRi for $i \in \{1, 2\}$. Definitions of the other operations can be found in [6] and are not given here, since Roach [7] has shown that there is a normal form for Head Grammars which only uses these operations.

$$\begin{array}{ll} LC1(u_1\overline{a_1}u_2, \overline{v}) = u_1\overline{a_1}u_2 v, & LC2(\overline{u}, v_1\overline{a_2}v_2) = u v_1\overline{a_2}v_2, \\ LL1(u_1\overline{a_1}u_2, \overline{v}) = u_1\overline{a_1}v u_2, & LL2(u_1\overline{a_1}u_2, \overline{v}) = u_1 a_1 \overline{v} u_2, \\ LR1(u_1\overline{a_1}u_2, \overline{v}) = u_1 v \overline{a_1} u_2, & LR2(u_1\overline{a_1}u_2, \overline{v}) = u_1 \overline{v} a_1 u_2 \end{array}$$

Both Pollard [6] and Roach [7] define these operations as partial functions. Pollard's definition of headed strings includes the headed empty string ($\overline{\lambda}$). However, mathematically, $\overline{\lambda}$ does not have the same status as other headed strings: for example, $LC1(\overline{\lambda}, \overline{w})$ is undefined. In general, 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 results about Head Grammars [7], and has caused problems in showing the equivalence of MHG's and HG's (see section 2).

The language generated by a HG G is defined as follows: $L(G) = \{ w \mid S \overset{*}{\Longrightarrow} \overline{w} \}$

Example 1.2 We now present a sample Head Grammar for the language $L_1 = \{ a^n g b^n f c^n h \mid n \geq 0 \}$

$$\begin{array}{ll} S \rightarrow LL2(S_1, \overline{f}), & S \rightarrow LL2(S_2, \overline{f}), \\ S_1 \rightarrow LC2(\overline{a}, S_3), & S_2 \rightarrow \overline{g}h, \\ S_3 \rightarrow LL2(S_1, \overline{b}c), & S_3 \rightarrow LL2(S_2, \overline{b}c) \end{array}$$

1.3. Modified Head Grammars

We find it convenient to consider a formalism that closely resembles HG's: referred to as Modified Head Grammars (MHG's). Instead of headed strings, MHG's have **split strings**. A split string has a distinguished position between two strings in V_T^*, about which it may be split. We will denote a split string as $w_1{\uparrow}w_2$ where $w_1 w_2 \in V_T^*$. Notice that we can represent the split empty string as $\lambda{\uparrow}\lambda$, though this will be denoted λ whenever the context makes it obvious that we are referring to a split string. In MHG's, there are 3 operations on split strings: W, $C1$, and $C2$, defined as follows:

$$\begin{aligned} W(w_1{\uparrow}w_2, u_1{\uparrow}u_2) &= w_1 u_1{\uparrow}u_2 w_2 \\ C1(w_1{\uparrow}w_2, u_1{\uparrow}u_2) &= w_1{\uparrow}w_2 u_1 u_2 \\ C2(w_1{\uparrow}w_2, u_1{\uparrow}u_2) &= w_1 w_2 u_1{\uparrow}u_2 \end{aligned}$$

The operations $C1$ and $C2$ correspond to the operations $LC1$ and $LC2$ in HG's. The operation W has been defined such that the split point of its second argument becomes the split point of the string resulting from application of the operation (like the HG operations $LL2$ and $LR2$).

Since the split point is not a symbol but a position between strings, separate operations corresponding to $LL2$ and $LR2$ are not needed. In addition, unlike HG's, which distinguish the two wrapping operations $LL1$ and $LL2$, W suffices as a substitute for both of these operations. Suppose $Y \overset{*}{\Longrightarrow} w_1{\uparrow}w_2$ and $Z \overset{*}{\Longrightarrow} u_1{\uparrow}u_2$ and we want X to derive $w_1{\uparrow}u_1 u_2 w_2$. This can be achieved with the following two productions: $Z^{fr} \rightarrow C1(\lambda, Z)$ and $X \rightarrow W(Y, Z^{fr})$.

Example 1.3 We now give a MHG generating L_1.

$$\begin{array}{ll} S \rightarrow W(S_1, f_{\uparrow}), & S \rightarrow W(S_2, f_{\uparrow}), \\ S_1 \rightarrow C2(a_{\uparrow}, S_3), & S_2 \rightarrow g_{\uparrow}h, \\ S_3 \rightarrow W(S_1, b_{\uparrow}c), & S_3 \rightarrow W(S_2, b_{\uparrow}c) \end{array}$$

We will defer the discussion of both the formal and linguistic relationship between HG's and MHG's until section 2. It is worth noting at this point that the definition of MHG's given here coincides with the definition of HG's given in Rounds [8]. As we shall see in section 2, these formalisms are very closely related.

1.4. Tree Adjunction and Wrapping

Before showing the formal equivalence of MHG's and TAG's, it is instructive to consider the relationship between the wrapping operation W of MHG's and the adjoining operation of TAG's. Suppose that we have the production $p = X \rightarrow W(Y, Z)$ in a MHG G, and that we have two derivations from the nonterminals Y and Z deriving the headed strings $w_1{\uparrow}w_2$ and $v_1{\uparrow}v_2$ respectively. Given the production p, we can derive the split string $w_1 v_1{\uparrow}v_2 w_2$ from X.

Suppose there is a derived auxiliary tree γ corresponding to the above derivation of $w_1{\uparrow}w_2$, from Y where the foot node appears at the split point, as shown in figure 1.6 below. Also assume that there is a node η dominating a subtree that corresponds to a derivation of $v_1{\uparrow}v_2$ from Z where, as before, we assume that the foot node appears at the split point. Consider the tree resulting from the adjunction of γ at the node η, also shown in figure 1.6. The resulting tree can be thought of as corresponding to the derivation of the split string $w_1 v_1{\uparrow}v_2 w_2$ from X.

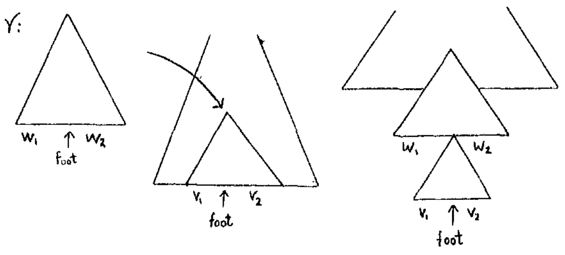


figure 1.6

This example illustrates the basic intuition behind the constructions involved in the following proofs showing the equivalence of MHG's and TAG's.

2. Head Grammars and Modified Head Grammars

In this section, we shall discuss the relationship between MHG's and HG's. First we present the outline of a construction showing that for every HG G there is an equivalent MHG G'. We then briefly discuss the linguistic relationship between MHG's and HG's.

Suppose $X \overset{*}{\Longrightarrow} w_1 \overline{h} w_2$. This headed string can be split in two ways: into the substrings w_1 and $h w_2$; or $w_1 h$ and w_2. This depends on whether X is used in a left or right wrapping operation. Since in MHG's we can only split a string in one place, we use two nonterminals, X^l and X^r deriving $w_1 h{\uparrow}w_2$ and $w_1{\uparrow}h w_2$ respectively. Thus, for example, the production $Z \rightarrow W(X^l, Y)$ can be used in place of $Z \rightarrow LL2(X, Y)$. A further complication arises when a headed string is split first to the right of its head and then the resulting string is split to the left of the same head. The problem is resolved by introducing 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. We can reintroduce the missing head with the following productions:

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

Complete details of this proof are given in [4].

We are unable to prove the inclusion of MHL's in HL's. The problems faced when attempting to find such a proof are a result of the operations in HG's not being total functions. For example, $C1(\overline{\lambda},\overline{w})$ is defined in MHG's, whereas $LC1(\overline{\lambda},\overline{w})$ is undefined in the HG's framework. We have not found any way of getting around this technical problem in a systematic manner. All TAG's considered by the authors so far have an equivalent HG. We feel that the problem of the empty headed string in the HG formalism does not result from an important difference between the formalisms.

In the following discussion, we propose that MHG's can be given a linguistic interpretation if we retain the notion of a head terminal in a split string. The split point should be viewed as determining the position of the head. As far as the authors are aware, Head Grammars for natural languages use only one kind of wrapping operation: either only the left wrapping operations LLi, or only the right wrapping operations LRi. Thus, any headed string can be split on only one side of the head. For example, if $w_1\overline{h}w_2$ is a headed string, and only the left wrapping operations were used, then the headed string can only be split as w_1h and w_2. For any HG using only left wrapping operations there exists an equivalent MHG such that split strings will have their split points immediately to the right of the actual head. However, obviously not every MHG can be given a linguistic interpretation in this way.

3. Equivalence of MHG's and TAG's

We will now show that the class MHL is equal to the class TAL. The complete proofs for the results presented here are given in [4].

3.1. Inclusion of TAL in MHL

Based on the earlier observation concerning the similarity between the wrapping and adjoining operations, we shall now present a scheme for transforming a given TAG $G = (V_N, V_T, S, I, A)$ to an equivalent MHG $G' = (V'_N, V_T, S, P)$. In this section, we have generalized the concatentation operations of MHG's to be of the form Cj for $j \geq 1$. It is obvious that these operations can always be simulated using just $C1$ and $C2$.

We shall first describe the algorithm *convert* informally. If η is a node of some elementary tree γ, applying *convert* to η returns a sequence of productions in the MHG formalism capturing the structure of the subtree of γ rooted at η. The wrapping operation is used to simulate the effect of adjunction; the concatenation operations Ci concatentate the strings derivable from the daughters of a node. The choice of i depends on which child is the ancestor of the foot node. The exact structure of a tree can be captured by using nonterminals that are named by the addresses of nodes of elementary trees rather than the nonterminals labelling the nodes.

The main idea of our scheme is as follows. Let $\langle\beta, i\rangle$ be the address of a node in an auxiliary tree β, and γ belongs to $\mathcal{D}(\langle\beta, i\rangle)$ with a frontier w_1Xw_2. We have a nonterminal corresponding to this node (denoted by $[\beta, i]$) which derives the split string $w_1{\uparrow}w_2$. In particular, when $\langle\beta, i\rangle$ is the root of β (i.e., $i = \epsilon$), then the nonterminal $[\beta, \epsilon]$ should derive the split strings $w_1{\uparrow}w_2$ whenever there is a tree in $\mathcal{D}(\beta)$ with frontier w_1Xw_2. That is, the split point appears in a position corresponding to the foot node.

Thus, the wrapping operation W can be used to simulate the effect of adjoining in the following manner. If $\langle\gamma, i\rangle$ is a node at which β is adjoinable, we have a production corresponding to adjunction of β at $\langle\gamma, i\rangle$.

$$[\gamma, i] \rightarrow W([\beta, \epsilon], \overline{[\gamma, i]})$$

where $\overline{[\gamma, i]}$ derives strings derivable from the children of $\langle\gamma, i\rangle$. We also have the rule $[\gamma, i] \rightarrow \overline{[\gamma, i]}$ for the case when no adjunction takes place at $\langle\gamma, i\rangle$. Since $\overline{[\gamma, i]}$ is supposed to derive strings derivable by the concatenation of the frontiers of subtrees dominated by the children of $\langle\gamma, i\rangle$, we have the production,

$$\overline{[\gamma, i]} \rightarrow Cj([\gamma, i\cdot 1], \ldots, [\gamma, i\cdot j], \ldots, [\gamma, i\cdot k])$$

where $\langle\gamma, i\cdot 1\rangle, \ldots, \langle\gamma, i\cdot j\rangle, \ldots, \langle\gamma, i\cdot k\rangle$ correspond to the k children of $\langle\gamma, i\rangle$ and where the j^{th} child is the ancestor of the foot node. The operation Cj is used so that the split point appears in the same position as the foot node. By convention, we let j be 1 when $\langle\gamma, i\rangle$ is not the ancestor of the foot node.

We are now in a position to define the conversion process. The algorithm is as follows:

```
for each initial tree $\alpha$,  let  $S \rightarrow [\alpha, \epsilon] \in P$.
for each elementary tree $\gamma$, call *convert*$(\langle\gamma, \epsilon\rangle)$
```

where the procedure *convert* is as defined below.

```
define *convert*$(\langle\gamma, i\rangle)$;
case 1: $\langle\gamma, i\rangle$ *is a leaf node*
    if $\langle\gamma, i\rangle$ has label $a \in V_T \cup \{\lambda\}$ then
        **step 1**: add $[\gamma, i] \rightarrow a_{\uparrow}$ to $P$
    else $[\gamma, i]$ *is the foot node*
        **step 2**: add $[\gamma, i] \rightarrow W([\beta, \epsilon], \overline{\lambda})$
            for each $\beta$ in SA constraint of $\langle\gamma, i\rangle$
        **step 3**: add $[\gamma, i] \rightarrow \overline{\lambda}$
            if $\langle\gamma, i\rangle$ does not have an OA constraint;
case 2: $\langle\gamma, i\rangle$ *is an internal node and has k children*
    **step 4**: add $[\gamma, i] \rightarrow W([\beta, \epsilon], \overline{[\gamma, i]})$
        for each $\beta$ in SA constraint of $\langle\gamma, i\rangle$
    if $\langle\gamma, i\rangle$ does not have OA constraint then
        **step 5**: add $[\gamma, i] \rightarrow \overline{[\gamma, i]}$
    if $\langle\gamma, i\rangle$ is ancestor of foot node then
        **step 6**: add $\overline{[\gamma, i]} \rightarrow Cj([\gamma, i\cdot 1], \ldots, [\gamma, i\cdot k])$
            where $j^{th}$ child dominates foot node;
    if $\langle\gamma, i\rangle$ is not ancestor of foot node then
        **step 7**: add $\overline{[\gamma, i]} \rightarrow C1([\gamma, i\cdot 1], \ldots, [\gamma, i\cdot k])$
    for $1 \leq j \leq k$ do *convert*$(\langle\gamma, i\cdot j\rangle)$.
```

We prove the inclusion of $L(G)$ in $L(G')$, by induction on the height of the trees derived from all subtrees of elementary trees, where the inductive hypothesis states:

For all elementary trees γ, and addresses i in γ, if there is a tree γ' in $\mathcal{D}(\langle\gamma, i\rangle)$ of height less than k, and the frontier of $\gamma' = w_1Xw_2$ or w_1w_2, then $[\gamma, i] \overset{*}{\Longrightarrow} w_1{\uparrow}w_2$.

It will be easy to show the inclusion of $L(G)$ in $L(G')$ by induction, considering steps 4, 5, 6 and 7. The base cases correspond to steps 1, 2 and 3.

We show the inclusion in the other direction by induction on the length of derivation of split strings in G'. The induction hypothesis is given by:

if $[\gamma, i] \overset{*}{\Longrightarrow} w_1{}_\uparrow w_2$ in k steps, then there is a $\gamma' \in \mathcal{D}(\langle\gamma, i\rangle)$ such that the frontier of γ' is $w_1 X w_2$ or $w_1 w_2$, depending on whether the foot node of γ, labelled by X if it exists, is a descendant of $\langle\gamma, i\rangle$ or not.

3.2. Inclusion of MHL in TAL

When we convert a TAG into a MHG, each elementary tree generates a set of productions. The sets generated by any two distinct elementary trees are disjoint and, furthermore, have a constrained form encoding the hierarchical structure of the tree. The task of converting a MHG to a TAG cannot simply involve the inversion of this construction since it is not in general possible to find groupings of productions in a MHG that have the required structure.

The approach used to convert MHG's to TAG's is based on satisfying the following requirement: for each 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: i.e., $X \overset{*}{\Longrightarrow} w_1{}_\uparrow w_2$ in MHG if and only if there is a derived auxiliary tree γ having no OA constraints, with root labelled X and frontier $w_1 X w_2$.

Suppose we had derived trees corresponding to derivations for B and C (as shown in the center of figure 3.1) that satisfied the above requirement.

We can capture the effect of each MHG production directly by associating exactly one elementary tree with each production. For example, figure 3.1 illustrates trees associated with the productions $A \to C1(B, C)$ (on the left) and $A \to W(B, C)$ (on the right). We position the foot node in the elementary trees to ensure that the split point and foot node appear at the same position. When the tree corresponding to wrapping is used the string derived from B is wrapped around the string derived from C. The foot of the resulting tree will appear immediately under the foot of the derived tree for C.

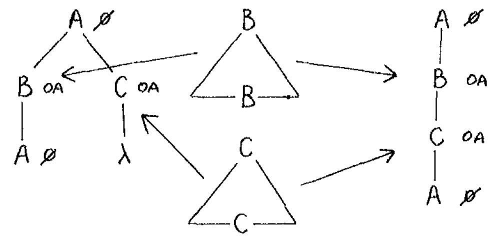


figure 3.1

The TAG that we produce could be viewed as simulating rewriting of nonterminals. Each rewriting corresponds to one use of the adjoining operation. The NA constraints at the root and the foot node of each auxiliary tree ensure that each occurence of a nonterminal is rewritten only once. The OA constraints are used to ensure that every nonterminal introduced is rewritten.

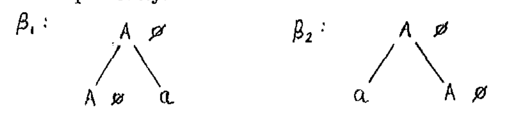
We now present the complete construction. Without loss of generality, we will assume a normal form that uses productions of the following form:

$$A \to f(B, C) \quad \text{or} \quad A \to {}_\uparrow a \quad \text{or} \quad A \to a_\uparrow$$

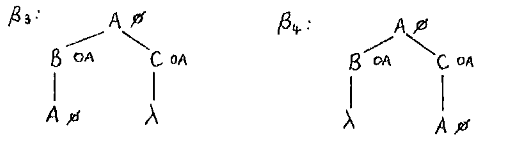
where $A, B, C \in V_N$, $a \in V_T \cup \{\lambda\}$ and $f \in \{C1, C2, W\}$.

The conversion proceeds as follows:

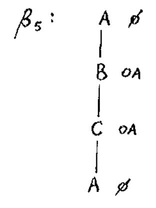
1. If $A \to {}_\uparrow a \in P$ or $A \to a_\uparrow \in P$ then include β_1 or β_2 in A respectively.



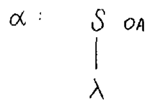
2. If $A \to C1(B, C) \in P$ or $A \to C2(B, C) \in P$ then include β_3 or β_4 in A respectively.



3. If $A \to W(B, C) \in P$ then include β_5.



The set I of initial trees consists of the single tree α:



We prove that $L(G) \subseteq L(G')$ by induction on the length of the MHG derivation. We show that if $X \overset{*}{\Longrightarrow} w_1{}_\uparrow w_2$ then there is a derived auxiliary tree having no OA constraints, with root labelled X and frontier $w_1 X w_2$.

We prove that $L(G') \subseteq L(G)$ by induction on the height of the TAG derivation tree. We show that if $\gamma \in \mathcal{D}(\beta)$, has no OA constraints, and has frontier $w_1 X w_2$ then $X \overset{*}{\Longrightarrow} w_1{}_\uparrow w_2$.

While straightforward, the proof given above does not capture the linguistic motivation underlying TAG's. The auxiliary trees directly reflect the use of the concatenation and the wrapping operations. It is also interesting to note that a consequence of the equivalence of MHG's and TAG's and the construction used in proving the inclusion of MHL in TAL is that we have the following normal form for TAG's. For any TAG there is an equivalent TAG with exactly one initial tree and auxiliary trees which are of five possible forms shown above.

4. Conclusion

In this paper we have shown the equivalence of TAL and MHL. Since we have also established the inclusion of HL in MHL we have shown that HLs are included in TALs. The inclusion of MHL in HL, and thus the equivalence of HG's and TAG's, in the most general case remains to be established We briefly discuss the relationship between MHG's and HG's and argue that it is close, both linguistically and formally. Figure 4.1 provides a summary of these results.

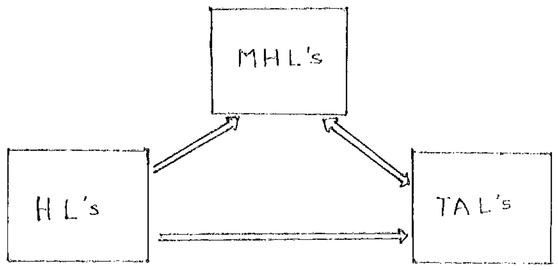


figure 4.1

References

[1] Gorn, S. 1962 Processors for Infinite Codes of the Shannon-Fano type, *Proc. of Symp. Math. Theory of Automata.*

[2] Joshi, A.K. 1983 How Much Context-Sensitivity is Necessary for Characterizing Structural Descriptions -- Tree Adjoining Grammars, in *Natural Language Processing -- Theoretical Computational and Psychological Perspectives,* eds. D. Dowty, L. Karttunen, A. Zwicky, Cambridge University Press, New York.

[3] Joshi, A.K. Levy, L. and Takahashi, M. 1975 Tree Adjunct Grammars. *Jour. Comp. Sys. Sci.* 10(1): 136-163.

[4] Joshi, A.K., Vijay-Shanker, K. and Weir, D.J. 1986 Tree Adjoining Grammars and Head Grammars, Tech. Rep., MS-CIS-86-1, Dept. of CIS, Univ. of Pennsylvania.

[5] Kroch, A. and Joshi, A.K. 1985 The Linguistic Relevance of Tree Adjoining Grammars, Tech. Rep., MS-CIS-85-18, Dept. of CIS, Univ. of Pennsylvania.

[6] Pollard,C. 1984 Head Grammars. Ph.D. dissertation, Standford Univ., California.

[7] Roach, K. 1984 Formal Properties of Head Grammars, Manuscript, Standford University. To appear in *Mathematics of Language,* ed. A. Manaster-Ramer, Amsterdam: John Benjamins.

[8] Rounds, W.C. 1985 LFP: A Logic for Linguistic Descriptions and an Analysis of its Complexity, Unpublished Manuscript, Univ. of Mich., Ann Arbor, Sept 1985.

[9] Vijay-Shanker, K. and Joshi, A.K. 1985 Some Computational Properties of Tree Adjoining Grammars. *Proc. of 23rd Meeting of Assoc. Comp. Ling.,* 82-93.

[10] Weir, D.J., Vijay-Shanker, K. and Joshi, A.K. 1986 The Relationship Between Tree Adjoining Grammars and Head Grammars, *Proc. of 24th Meeting of Assoc. Comp. Ling.*