Is Adjunction Compositional?

Marcus Kracht
Department of Linguistics, UCLA
PO Box 951543
405 Hilgard Avenue
Los Angeles, CA 90095–1543
USA
kracht@humnet.ucla.de

Abstract

The idea of adjunction can be traced back to the beginnings of generative syntax. Already Zellig Harris used adjunction and in [Harris, 1979] even exposes the idea that sentence meanings can be derived from kernel sentences via adjunction. Today, TAGs and the somewhat lesser known contextual grammars ([Marcus, 1967]) still use adjunction as a major syntactic operation. Recently, [Kallmeyer and Joshi, 2003] have provided a semantics for TAGs they claim to be compositional. However, this semantics contains a partial record of the derivation, thus adding detail that is of no semantic relevance.

1 Introduction

Recently, with the appearance of Hodges seminal work [Hodges, 2001], interest in compositionality has come back into focus. Yet, as much as this is a good development, what is still missing is a thorough discussion of what linguistic frameworks are actually compositional and which ones are not. Whether or not a given syntactic analysis is 'compositional' is still a matter of personal conviction and taste.

Given that TAGs are considered an alternative to the prevailing transformational account (see [Frank, 2002]) it is crucial to see how well TAGs are suited for the purposes of semantic interpretation. In this paper I shall show that the situation is not so favourable at all. In my view, TAGs cannot provide an analysis of natural languages in a compositional way. I propose a bundle of arguments to support this claim. It follows that the question of compositionality is not vacuous despite claims to the contrary (see [Zadrozny, 1994]): there are syntactic frameworks which actually are not compositional. This may come as a surprise because it seems unclear what the right kind of semantics is. In fact, [Kallmeyer and Joshi, 2003] have recently proposed a semantics for TAGs which they claim is compositional. So how do we decide?

Below we shall give an argument that is not based on any concrete semantics but instead uses the notion of synonymy. It assumes that certain sentences are synonymous while others are not. If one agrees with my claims concerning synonymy it immediately follows that since [Kallmeyer and Joshi, 2003] succeed in computing meanings of composite expressions, they must attribute distinct meanings to synonymous items. This is quite a common situation in theoretical semantics. Many approaches to semantics do not raise the question whether what they call semantics actually only consists in the meaning. A particularly striking example is the paper [Zadrozny, 1994], which argues that any language is compositional; the proof proceeds by replacing the original meanings by functions on meanings. Clearly, if the notion of compositionality is to make any sense at all, we are not free to distinguish synonymous items by means of any device whatsoever.

2 The Main Argument

A Tree Adjunction Grammar knows two kinds of trees: complete trees and adjunction trees. A **complete tree** is one in which all leaves end in a terminal symbol; an **adjunction tree** is a tree in which all but one leaf is labeled by a terminal symbol and the nonterminal label of that leaf is identical to the root. Adjunction works as follows. If a tree T contains a node u with label X and there is an adjunction tree U whose root (and designated leaf) has label X, then we bisect the tree T into the constituent C headed by u and the remainder. We insert the adjunction tree U between C and the remainder. (See Section 6 below, or [Frank, 2002], [Kracht, 2003] or [Kallmeyer and Joshi, 2003] for details.) A TAG is (adjunction restrictions aside) a pair consisting of a finite set of complete trees (called **elementary trees**) and a finite set of adjunction trees (called **auxiliary trees**). The language of trees is the one generated from the elementary trees by adjunction of auxiliary trees. Trees are assumed to be binary branching, so that every node can be named by a binary sequence, called its **address**. The operation of adjunction uses as input three givens: a complete tree T, an auxiliary tree U, and the address α of u. We denote the result by $T_\alpha(U, T)$. We shall supply more details below in Section 6.

Clearly, we expect the following from a compositional grammar. If two syntactic structures are synonymous and we apply the same syntactic operation to them, the results must also be synonymous. Now here is the example where adjunction fails to preserve synonymy. Consider a grammar which has the the trees of Figure 1. This grammar generates the following language:

(1)
$$\begin{aligned}&\{\texttt{a drunken}^n\ \texttt{sailor and a drunken}^m\ \texttt{captain were caught}\\ &\qquad\texttt{speeding} : m, n \in \mathbb{N}\}\\ &\cup\{\texttt{a drunken}^n\ \texttt{captain and a drunken}^m\ \texttt{sailor were caught}\\ &\qquad\texttt{speeding} : m, n \in \mathbb{N}\}\end{aligned}$$

Concerning semantics, I only assume that `and` is commutative, so that $\vec{x}$ `and` $\vec{y}$ and $\vec{y}$ `and` $\vec{x}$ are synonymous. Here is now the dilemma. Write $\alpha \approx \alpha'$ if α and α' are trees with synonymous yield. Then we have:

(2)
$$\sigma \approx \tau, \qquad T_{01}(\delta, \sigma) \not\approx T_{01}(\delta, \tau)$$

This is because the following two sentences are the yield of $T_{01}(\delta, \sigma)$ and $T_{01}(\delta, \tau)$, respectively:

(3) `A drunken sailor and a captain were caught speeding.`

(4) `A drunken captain and a sailor were caught speeding.`

The two are clearly not synonymous.

This argument is vulnerable to the claim that the two sentences actually do not have the same syntactic structure. This position is viable but meets insurmountable difficulties. We can, for example, construct a sentence that involves any number of NPs, such as

(5) `A sailor, a barman, a police officer and a captain were caught speeding.`

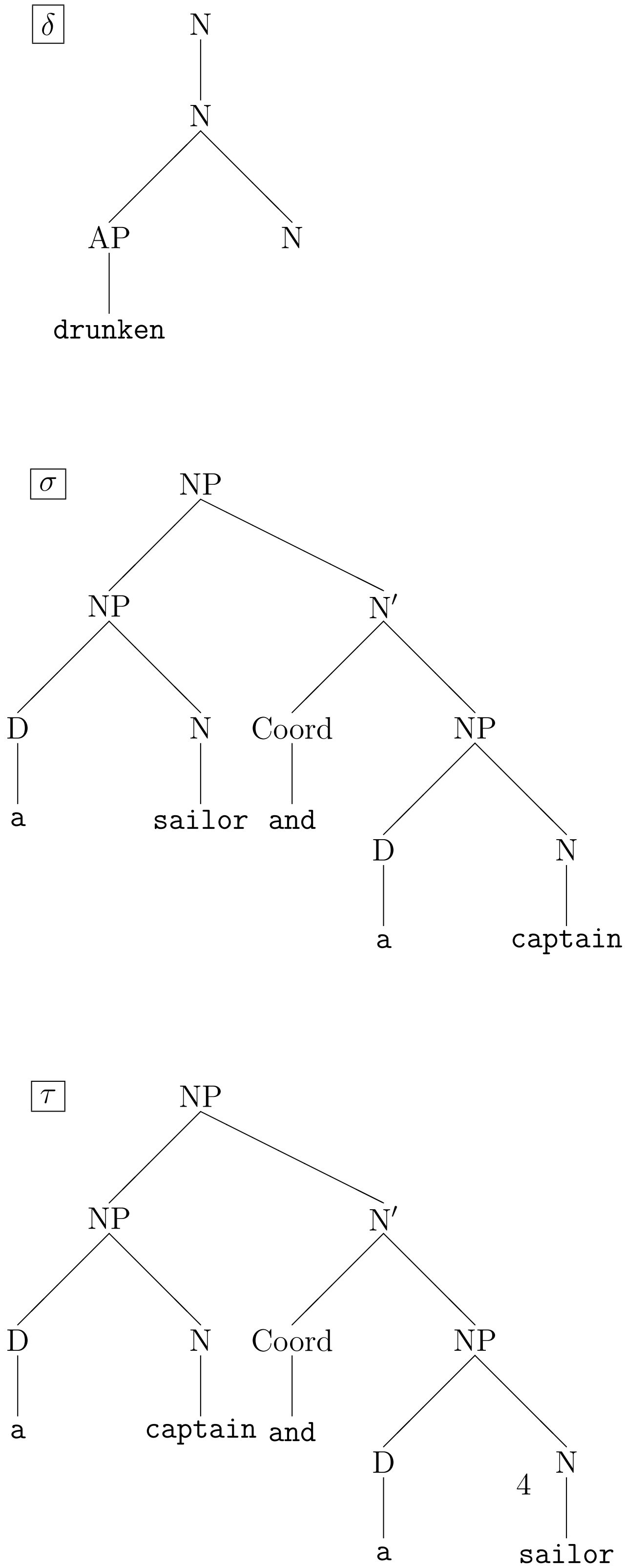

Figure 1: Adjunction Tree and Centre Trees

This sentence has 24 synonymous alternatives obtained by permuting the conjuncts. Adjoining `drunken` to the first we obtain six different sentences, each with six synonymous permutations.

(6) `A drunken sailor, a barman, a police officer and a captain were caught speeding.`

(7) `A drunken sailor, a police officer, a barman and a captain were caught speeding.`

(8) `A drunken sailor, a captain, a police officer and a barman were caught speeding.`

(9) `A drunken sailor, a police officer, a captain and a barman were caught speeding.`

(10) `A drunken sailor, a captain, a barman and a police officer were caught speeding.`

(11) `A drunken sailor, a barman, a captain and a police officer were caught speeding.`

At each of these places we may adjoin some other tree and further reduce synonymy. If one wishes to maintain the claim that all this is due to different syntactic structure, one is forced to believe the following. In a conjunction of n NPs, in any two permutations that have two different jth conjuncts C and D, either the category of C is different from that of D, or the address of C is different from Ds address. Additionally, for large enough n, the coordinate structures must have been built through the use of adjunction. It can be shown that they must have been built from some similar structure with less coordinated members. As there are only finitely many auxiliary trees, one can find a pair C and D which has been introduced by adjoining the same auxiliary tree Y. If adjunction of the adjective (or any other auxiliary tree V) on a given node in Y is licit in one structure, it is licit in all. Hence, on this pair we actually have an instance of an adjunction of the same auxiliary tree V at the place of C and D. Still, the possibility remains that C and D, although both conjuncts number j, actually have different address. This eventually leads to the claim that in all permutations of a coordinated structure, the analogous conjuncts have different address unless literally identical.

At this point one may think the conditions are impossible to meet. This is not so. In fact, I believe that it is possible to structurally encode the semantics in such a way that compositional interpretation is possible. However, it consists in giving up any homogeneous account of syntactic structure whatsoever. To see what it takes, let me point out that if two nouns, say `sailor` and `captain` receive the same syntactic category in a grammar, they are interchangeable in all strings. This is so for TAGs, CFGs, and so on. In fact, one expects the converse to hold as well. If two strings are intersubstitutable grammar should not distinguish them. We shall explicify this principle. Languages are taken to be strings of letters, words are strings enclosed by a blank. Thus, a sentence is a string of words. We assume that trees for a language L are trees whose terminal leaves are labeled by words.

Definition 1 *Let L be a language. Two strings $\vec{x}$ and $\vec{y}$ are called L–**indistinguishable** if for every strings $\vec{u}$ and $\vec{v}$: $\vec{u}^\frown\Box^\frown\vec{x}^\frown\Box^\frown\vec{v} \in L$ if and only if $\vec{u}^\frown\Box^\frown\vec{y}^\frown\Box^\frown\vec{v} \in L$. Let T be set of trees language for L. T is **syntactically adequate for** L if whenever $\vec{x}$ and $\vec{y}$ are indistinguishable in L, then substituting $\vec{x}$ for $\vec{y}$ at an address α in a tree of T also yields a tree of T.*

The principle is now this.

> IDENTITY OF INDISTINGUISHABLES. Any tree set T for L must be syntactically adequate.

On the basis of this requirement it suffices to find two nouns which are indistinguishable in English.

If one is not prepared to grant the existence of two nouns, or wishes to deny the above principle, there is a more devastating attack, which I shall launch below in Section 6. To put matters simply: the address of adjunction is a syntactic device with no equivalent in semantics. Its identity cannot be communicated to semantics in a compositional grammar. It follows that the modes of composition cannot even be defined for TAGs.

3 Taking A Closer Look

Let us now take a look at the semantic representations and see how the failure of compositionality shows up there. I assume that in nominal coordination, `and` forms a set. (In fact, this assumption is not needed. What is needed is only that the meaning of `and` is commutative.) The representation of

(12) `A sailor and a captain were caught speeding.`

will be like this:

(13)

$X\ x_0\ x_1$
$\mathsf{caught{-}speeding}'(X);$ $X = \{x_0, x_1\}; \mathsf{captain}'(x_0);$ $\mathsf{sailor}'(x_1).$

Let us ask first: what does synonymy mean for DRSs? It means identity of truth conditions. The truth condition of (13) is that of

(14) $$(\exists X)(\exists x_0)(\exists x_1)(X = \{x_0, x_1\} \wedge \mathsf{sailor}'(x_1) \wedge \mathsf{captain}'(x_0) \wedge \mathsf{caught\text{-}speeding}'(X))$$

Hence, any suitable renaming of the variables yields a synonymous DRS. In particular, the same DRS is assigned to

(15) `A captain and a sailor were caught speeding.`

Now, the problem that we are facing is that we want to go from (13) to

(16)

$X\ x_0\ x_1$
$\mathsf{caught{-}speeding}'(X);$ $\mathsf{captain}'(x_0); \mathsf{drunken}'(x_0);$ $X = \{x_0, x_1\}; \mathsf{sailor}'(x_1).$

as well as to

(17)

$X\ x_0\ x_1$
$\mathsf{caught{-}speeding}'(X);$ $\mathsf{captain}'(x_1); \mathsf{drunken}'(x_0);$ $X = \{x_0, x_1\}; \mathsf{sailor}'(x_0).$

There is a temptation here that must be avoided. Suppose we give (15) the representation

(18)

$X\ x_0\ x_1$
$\mathsf{caught{-}speeding}'(X);$ $X = \{x_0, x_1\}; \mathsf{captain}'(x_1);$ $\mathsf{sailor}'(x_0).$

Then we could simply say that the meaning of the adjoined tree is $\mathsf{drunken}'(x_0)$, and that it is inserted anywhere in the main DRS. So, we go from (13) to (16) and from (18) to (17). This solution however rests on a misunderstanding. The variables of DRSs are bound and can be renamed. As (13) and (18) are synonymous (but different), they are not the meanings, they are just proxy for the meanings. A solution based on meanings as such must be immune to renaming.

4 An Interesting Proposal

There is another possibility that may come to mind. We could make the indexation relative to the meaning of the noun. The DRS for σ is of this form:

$$(19) \qquad \begin{array}{|c|} \hline S\ x_0\ x_1 \\ \hline \Sigma \\ \hline \end{array}$$

The two individuals talked about in (12) can be distinguished by some property $\varphi(x)$ (here, one is a sailor and the other is a captain). If this property does not exist, then the descriptions offered by the nouns is identical. This case is unproblematic. Now, assume that the DRS Δ_1 (or whatever representation is actually chosen) for α_1 is arranged so that the individual that gets modified is denoted by x_0; similarly for the DRS Δ_2 of α_2. Denote by $[x_1/x_0]\Delta$ the result of uniformly replacing occurrences of x_0 in Δ by x_0 (and doing some other renamings if necessary, that is, if x_1 already occurs in Δ). Now, define an operation, A, which works as follows:

$$(20) \qquad A(\Delta_1, \Delta_2) := \begin{cases} \begin{array}{|c|} \hline S\ x_0\ x_1 \\ \hline \Sigma; \Delta_1; [x_1/x_0]\Delta_2 \\ \hline \end{array} & \text{if } \Sigma \vDash \mathsf{sailor}'(x_0) \\ \\ \begin{array}{|c|} \hline S\ x_0\ x_1 \\ \hline \Sigma; [x_1/x_0]\Delta_1; \Delta_2 \\ \hline \end{array} & \text{if } \Sigma \nvDash \mathsf{sailor}'(x_0) \end{cases}$$

To understand the reason for this definition recall that it is independent of the way the variables are named; we could exchange x_0 and x_1 in Δ and the result is still well–defined. One must remember that the definition needs to be independent of what the variables are actually called. (Obviously, if the variables can be something else but x_0 and x_1, the definition must be changed accordingly.)

This operation modifies the variable satisfying φ, which by definition is the one object whose description is to the left in (12) and to the right in (15). Now, the semantic function accompanying $T_{01}(\alpha_1, \alpha_2)$ is $A(\Delta_1, \Delta_2)$. This unfortunately gives the correct result only for (12).

Thus, in order to distinguish (12) and (15), we simply need to inhibit adjunction for (15). We have argued against that solution above, but let us see where this proposals leads us to. We introduce another function $T^{\bullet}_{01}(\alpha_1, \alpha_2)$, whose result is the same as $T_{01}(\alpha_2, \alpha_1)$, but the accompanying semantics is identical to $A(\Delta_1, \Delta_2)$. Similarly, we need to make sure that $T^{\bullet}_{01}(\alpha_1, \alpha_2)$ is undefined. So, we double the centre trees σ and τ; one of the versions, $\sigma_\circ$ and $\tau_\circ$, bans the adjunction of δ_1 to the higher node and the other, $\sigma_\bullet$ and $\tau_\bullet$, bans the adjunction of τ to the lower node. A lot of details still need to be arranged. But the basic construction should be clear.

The problem with this approach is that the substitution function depends on the actual semantics of the noun that one adjoins to. This works in the present grammar, but in general there are infinitely many noun denotations.

5 Semantic Failures of Compositionality

The argument I have presented above may not appeal to everyone, since it rests on the assumption that there exists synonymous sentences of particular kind plus two L–indistinguishable nouns. However, if one is unwilling to grant me this much, still I think there are good reasons why TAGs are just not compositional.

Assume that sentences denote functions from possible worlds to truth values, given by formulae. The meaning of the sentence

(21) `A man sees a dog.`

is (give or take changing the order of the conjuncts) given by the formula

(22) $(\exists x)(\exists y)(\mathsf{man}'(x) \wedge \mathsf{dog}'(y) \wedge \mathsf{see}'(x, y))$

We want to insert the phrase $\lambda\mathcal{P}.\lambda x.\mathsf{big}'(x) \wedge \mathcal{P}(x)$ in order to get either of the following.

(23) $(\exists x)(\exists y)(\mathsf{big}'(x) \wedge \mathsf{man}'(x) \wedge \mathsf{dog}'(y) \wedge \mathsf{see}'(x, y))$

(24) $(\exists x)(\exists y)(\mathsf{man}'(x) \wedge \mathsf{big}'(y) \wedge \mathsf{dog}'(y) \wedge \mathsf{see}'(x, y))$

The order of conjuncts is again of no particular concern here. The problem is, so to speak, to insert the adjective 'post facto' into the derivation. But there is no derivation to begin with. What is at stake is that the meaning of (21) cannot be generated from (22). The reason is that we need to know exactly what variable quantifies the expression for `dog`, or which one corresponds to `man`. But to spy into a quantifier to see what variable it binds is to deny that changing bound variables is an operation that leaves meanings invariant! (Even worse: for adherents of Frege, the meaning of an extensional sentence is a truth value, so every pair of true sentences has the same meaning.) The same problem arises with DRSs, which use open formulae, as in [Kamp and Reyle, 1993]. Suppose that we want to adjoin the tree for `big`, we need to assume that the is an argument available to which `big` can be applied. In ordinary DRSs, these would be the discourse markers for the objects corresponding to `man` and `dog`. Let them be x and y. For the adjunction, we want to proceed from

(25) $$\begin{array}{|c|}\hline x\ y \\ \hline \mathsf{man}'(x); \mathsf{dog}'(y); \\ \mathsf{see}'(x,y). \\ \hline \end{array}$$

to either of the following two:

(26) $$\begin{array}{|c|}\hline x\ y \\ \hline \mathsf{man}'(x); \mathsf{dog}'(y); \\ \mathsf{see}'(x,y); \mathsf{big}'(x). \\ \hline \end{array} \qquad \begin{array}{|c|}\hline x\ y \\ \hline \mathsf{man}'(x); \mathsf{dog}'(y); \\ \mathsf{see}'(x,y); \mathsf{big}'(y). \\ \hline \end{array}$$

Once again, the adjunction is successful only if it is known prior to its execution whether we have chosen the right variable. Notice that the choice is not just between x and y. Kamp and Reyle as well as Joshi and Kallmeyer resort to syntactic indices to get the insertion of variables right. The problem is much deeper than ordinarily assumed; it has been made forcefully in [Vermeulen, 1995]. The phrase `a man` is made from two DRSs, one for `a` and one for `man`. However, to be really exact, the DRSs are like this, for each possible choice of $i, j \in \mathbb{N}$:

(27) $$\begin{array}{c} \mathsf{a}_i \\ \begin{array}{|c|}\hline x_i \\ \hline \varnothing \\ \hline \end{array} \end{array} \qquad \begin{array}{c} \mathtt{man}_j \\ \begin{array}{|c|}\hline \ \\ \hline \mathsf{man}'(x_j) \\ \hline \end{array} \end{array}$$

The translation algorithm takes annotated syntactic representations as input, annotation being done by numbers:

(28) $\texttt{A}_1$ $\texttt{man}_1$ $\texttt{sees}_{\langle 1,2\rangle}$ $\texttt{a}_2$ $\texttt{dog}_2$.

The index i is translated into the variable x_i. Correct indexation is a prerequisite for correct interpretation.

If adjunction is to work together with DRS formation, when adjoining a structure with another we need to make sure that the correct indices are chosen. However, one of the structures is of arbitrary size and its interpretation contains any number of indices, so it is not clear how it is ensured that the correct index is chosen when adjunction happens. If only one variable needs to be shared across DRSs, or if that number is bounded, it is conceivable that a naming convention for discourse markers can achieve the goal in question. For example, suppose adjunction gives rise to the identification of just one discourse marker. Then we can assume that the marker of the adjoining tree is uniformly x_0, the markers of the tree to which we adjoin x_0, x_1, ..., x_{n-1}, where n is less than a fixed number (it depends only on the grammar chosen). Then there are in principle only n ways to identify x_0. Assume that x_0 of the adjoining tree is identified with x_j of the other tree. Then the markers of the adjoining tree are renamed from x_i to x_{2i}, the markers from the other tree from x_j to x_{2j+1}, except for x_j, which receives the name x_0. Finally, the DRSs are merged.

The previous discussion suggests that adjunction can be given a compositional interpretation, if only on the representations. However, this depends on the availability of enough discourse markers to begin with. For example, there are adjectives that are not intersective but which depend on the description being used for the object. Examples are `big`, `tall`, `good`, and `fake`. If I say that John is a tall basketball player, probably I want to say that he is tall qua being a basketball player, not just tall as a grown up. Thus, `tall basketball player` takes the latter in its scope. Let our first sentence be

(29) `A policeman chases a basketball player.`

From this sentence we want to get to

(30) `A policeman chases a tall basketball player.`

Translated into DRSs we want to go from left to right:

(31)

x_0 x_1
policeman$'(x_1)$; basketball-player$'(x_0)$; chase$'(x_1, x_0)$.

x_0 x_1
policeman$'(x_1)$; tall$'$(basketball-player$'$)(x_0); chase$'(x_1, x_0)$.

Now it is less likely that such an operation exists, since it needs to decompose an expression in the DRSs, just like the first example above. This time, however, the expression is not identifiable by an index any more.

6 String Adjunction and Tree Adjunction

The rest of the paper is devoted to the following question: what needs to be changed in the definition of TAGs so as to make adjunction compositional? To anticipate the answer: in order to make TAGs compositional one needs to remodel the derivations in such a way that we get a Linear Context Free Rewrite System. There are two choices; the first construction ends in a LCFRS that uses adjunction, called adjunction grammar; this grammar is still vulnerable to the criticism of the earlier sections. However, it is the only way in which adjunction can be retained in a compositional grammar.

Strings are denoted by vector arrows. ε denotes the empty string. Concatenation is sometimes written $\vec{x}^\frown\vec{y}$, but usually $^\frown$ is omitted. The length of a string $\vec{x}$ is denoted by $|\vec{x}|$. A **context** is a pair of strings. If $C = \langle\vec{u}_1, \vec{u}_2\rangle$ is a context, we write

(32) $C(\vec{x}) := \vec{u}_1\vec{x}\vec{u}_2$

If $D = \langle\vec{v}_1, \vec{v}_2\rangle$ is another context, put

(33) $C(D) := \langle\vec{u}_1\vec{v}_1, \vec{v}_2\vec{u}_2\rangle$

Let $\vec{x}$ be a string, and let it be decomposed into

(34) $\vec{x} = \vec{u}_1\vec{v}\vec{u}_2$

Let $D = \langle\vec{y}, \vec{z}\rangle$ be a context. Then the **string adjunction** to $\vec{x}$ with respect to this decomposition is

(35) $C(D(\vec{v})) = \vec{u}_1\vec{y}\vec{v}\vec{z}\vec{u}_2$

Evidently, this operation is not uniquely defined by $\vec{x}$.

Adjunction of trees is defined as follows. We call a **tree** a pair $\mathfrak{T} = \langle T, \ell\rangle$, where T is a tree domain, and $\ell : T \to K$ a function, the labeling. Let tree $\mathfrak{U} = \langle U, m\rangle$ be given, and $\vec{x} \in T$, $\vec{y} \in U$, such that $m(\vec{y}) = m(\varepsilon) = \ell(\vec{x})$. Then the following operation is defined:

$$\begin{aligned}(36)\qquad \xi_{\vec{x},\vec{y}}(U, T) :=\ & \{\vec{z} : \vec{z} \in T, \vec{x} \text{ not a prefix of } \vec{z}\}\\ & \cup \{\vec{x}\vec{z} : \vec{z} \in U\}\\ & \cup \{\vec{x}\vec{y}\vec{v} : \vec{x}\vec{v} \in T\}\end{aligned}$$

$$\begin{aligned}(37)\qquad \xi_{\vec{x},\vec{y}}(\ell, m) :=\ & \{\langle\vec{z}, \ell(\vec{z})\rangle : \vec{z} \in T, \vec{x} \text{ not a prefix of } \vec{z}\}\\ & \cup \{\langle\vec{x}\vec{z}, m(\vec{z})\rangle : \vec{z} \in U\}\\ & \cup \{\langle\vec{x}\vec{y}\vec{z}, \ell(\vec{x}\vec{z})\rangle : \vec{x}\vec{z} \in T\}\end{aligned}$$

$$(38)\qquad \xi_{\vec{x},\vec{y}}(\mathfrak{T}, \mathfrak{U}) := \langle\xi_{\vec{x},\vec{y}}(T, U), \xi_{\vec{x},\vec{y}}(\ell, m)\rangle$$

$\vec{x}$ is called the **adjunction site**. The general assumption in TAGs is the following. A **centre tree** is a tree whose yield is a terminal string. An **adjunction tree** is a tree whose yield has the form $\vec{x}^\frown Y^\frown\vec{y}$, where Y is a nonterminal, and $\vec{x}$ and $\vec{y}$ are terminal strings. There is thus a single leaf whose leaf is a nonterminal symbol which we call **distinguished**. (The distinguished leaf is required to have the same label as the root and the adjunction site.) Under these assumptions, we agree on the following.

$$(39)\qquad \xi_{\vec{x}}(\mathfrak{U}, \mathfrak{T}) := \xi_{\vec{x},\vec{y}}(\mathfrak{U}, \mathfrak{T})$$

where $\vec{y}$ is the unique address of the distinguished leaf of $\mathfrak{U}$. Notice, however, that the adjunction site $\vec{x}$ must still be given.

Denote by $Y(\mathfrak{T})$ the yield of $\mathfrak{T}$. Let an address $\vec{c}$ be given. Then the nodes below $\vec{c}$ form a tree with yield $\vec{x}$. The context of this string is determined by $\vec{c}$. Let it be $C = \langle\vec{v}_1, \vec{v}_2\rangle$. Let $Y(\mathfrak{U}) = D(N) = \vec{u}_1 N\vec{u}_2$ for some nonterminal N. Pick an address $\vec{a}$ in $\mathfrak{T}$. Suppose we adjoin $\mathfrak{U}$ at $\vec{a}$. Now $\mathfrak{T}$ is factored into $C(\vec{x}) = \vec{v}_1\vec{x}\vec{u}_2$ by $\vec{a}$. Then

$$(40)\qquad Y(\xi_{\vec{a}}(\mathfrak{U}, \mathfrak{T})) = C(D(\vec{x})) = \vec{v}_1\vec{u}_1\vec{x}\vec{u}_2\vec{v}_2$$

Now, suppose we allow adjunction to adjunction trees. Then a few more cases need to be considered. Let $\vec{c}$ be the address of the distinguished leaf of $\mathfrak{T}$. An adjunction site $\vec{a}$ induces a decomposition of the yield of $\mathfrak{T}$.

① $\vec{a}$ is a prefix of $\vec{c}$. Then $Y(\mathfrak{T}) = \vec{v}_1\vec{u}_1 N\vec{u}_2\vec{v}_2$, and the constituent at $\vec{a}$ is $\vec{u}_1 N\vec{u}_1$.

② $\vec{a}$ precedes $\vec{c}$. Then $Y(\mathfrak{T}) = \vec{v}_1\vec{u}_1\vec{u}_2N\vec{v}_2$, and the constituent of $\vec{a}$ is $\vec{u}_1$

③ $\vec{b}$ precedes $\vec{c}$. Then $Y(\mathfrak{T}) = \vec{v}_1N\vec{u}_1\vec{u}_2\vec{v}_2$ and the constituent at $\vec{a}$ is $\vec{u}_2$.

Then

$$(41) \qquad Y(\xi_{\vec{a}}(\mathfrak{U}, \mathfrak{T})) = \begin{cases} \vec{v}_1\vec{u}_1\vec{x}_1N\vec{x}_2\vec{u}_2\vec{v}_2 & \text{in Case 1} \\ \vec{v}_1\vec{x}_1\vec{u}_1\vec{x}_2\vec{u}_2N\vec{v}_2 & \text{in Case 2} \\ \vec{v}_1N\vec{u}_1\vec{x}_1\vec{u}_2\vec{v}_2 & \text{in Case 3} \end{cases}$$

Now, the nonterminal defines the split into left and right part of the context. This is all that needs to be known to correctly compute the string adjunction. Therefore, the trees are not needed at all. To know whether adjunction is licit, we only need to know which category the adjunction site has.

A TAG is a pair consisting of a finite set of centre trees and a finite set of adjunction trees. In principle, adjunction trees can be adjoined if and only if adjunction is defined. TAGs allow in addition to restrict adjunction in the following way. The label of each node is a pair $\langle N, \mathbb{A}\rangle$, where N is a label of any sort, and $\mathbb{A}$ a set of adjunction trees. Adjunction of $\mathfrak{U}$ at $\vec{x}$ in a tree is possible only if the label is of the form $\langle N, \mathbb{A}\rangle$ with $\mathfrak{U} \in \mathbb{A}$. (The root and distinguished leaf of $\mathfrak{U}$ shall require special treatment; their label is inherited from $\mathfrak{T}$. Identity is not necessary for adjunction to be defined: only the first in the pair needs to match!)

7 Compositionality

We shall use the terminology of [Kracht, 2003]. A sign is a triple $\langle e, c, m\rangle$ consisting of an **exponent** e, a category c and a meaning m. A **language** is a set of signs. A **grammar** over $E \times C \times M$ consists of is a finite set F of function symbols, a signature $\Omega : F \to \mathbb{N}$, and for each $f \in F$, functions $f^\varepsilon : E^{\Omega(f)} \to E$, $f^\gamma : C^{\Omega(f)} \to C$ and $f^\mu : M^{\Omega(f)} \to M$ such that $\mathfrak{E} = \langle E, \{f^\varepsilon : f \in F\}\rangle$, $\mathfrak{C} = \langle C, \{f^\gamma : f \in F\}\rangle$, and $\mathfrak{M} = \langle M, \{f^\mu : f \in F\}\rangle$ are partial algebras. The signature defines the terms, and if $\mathfrak{t}$ is a constant term (a term without any variables), and if $\mathfrak{t}^\varepsilon$, $\mathfrak{t}^\gamma$ and $\mathfrak{t}^\mu$ all exist, then t is called **definite**. The **language generated by** G is the set of all $\langle \mathfrak{t}^\varepsilon, \mathfrak{t}^\gamma, \mathfrak{t}^\mu\rangle$ where $\mathfrak{t}$ is definite.

We shall point out a few consequences of the definitions. First, we have to choose what the exponents are and what the categories. In TAGs there is no split between the two: trees contain structural information together with labels of terminal nodes; especially in L-TAGs this is made a necessary mix. One can say that the trees the grammars operate on are the exponents. The categories are then not needed, but to conform to the format, we shall make the algebra trivial (containing a single member). Thus, we are essentially manipulating pairs of trees and meanings. It is not necessary to know what the meanings actually are. However, one thing is clear: tree adjunction is not uniquely defined by the trees alone. This is the same with strings. For example, in the sentence

(42) `The postman is biting the dog.`

we can adjoin the tree for adjective `ferocious` at two positions. We may think of the adjective as a context of the form $\langle \texttt{ferocious}, \varepsilon \rangle$, adjoined at two positions:

(43) `The` $\triangledown$ `postman` $\blacktriangle$ `is biting the dog.`

(44) `The postman is biting the` $\triangledown$ `dog` $\blacktriangle$`.`

This gives either

(45) `The ferocious postman is biting the dog.`

or

(46) `The postman is biting the ferocious dog.`

If we now turn to TAGs, we see that the trees for (43) and (44) are the same. The result is different because a different adjunction site has been chosen. But the resulting meanings are different. Thus, naming the adjunction site is necessary for compositionality.

This creates a problem. The more trees get adjoined to a tree, the larger the tree and the more adjunction sites may exist. This means that if just the two trees are known the result of adjunction is still undefined. There is no a priori limit on the number of possible adjunction sites. Thus we need to eliminate the need to name the adjunction site; on the other hand, the semantic functions need to be informed about the adjunction site. Since the only communication channel that exists between syntax and semantics is the choice of the function $f \in F$, and since F is finite, only a finite amount of information can be shared between syntax and semantics. If there is an infinity of adjunction sites, adjunction cannot be defined on the semantics. That there really are arbitrarily many adjunction sites in a sentence, follows easily by modifying our argument above.

We can eliminate the problem as follows. Instead of adjoining only to centre trees, we adjoin to either adjunction trees or centre trees, but on condition that they are basic. Since the latter are of bounded size, we can assume for each address $\vec{x} \in \mathbb{N}^*$ a different adjunction function $\xi_{\vec{x}}$. Notice however that a centre tree can have more than one adjunction site. If the adjunctions are done in sequence, only the first adjunction is to a basic tree of the grammar. So only the first is guaranteed to apply to a tree of bounded size. Thus, we need to reconsider adjunction, and allow for simultaneous adjunction of several adjunction trees at several sites. Thus, we are proposing to use multicomponent adjunction.

Let us summarize: to make TAGs compositional the basic operations must be simultaneous adjunctions to well–defined positions in a tree, and the tree adjoined to is restricted to a basic centre or a basic adjunction tree. Now, let $*$ be a binary operation on contexts and strings, which is defined as follows.

(47)
$$\begin{aligned} \langle \vec{u}_1, \vec{u}_2 \rangle * \vec{x} &:= \vec{u}_1 \vec{x} \vec{u}_2 \\ \langle \vec{u}_1, \vec{u}_2 \rangle * \langle \vec{v}_1, \vec{v}_2 \rangle &:= \langle \vec{u}_1 \vec{v}_1, \vec{v}_1 \vec{u}_2 \rangle \end{aligned}$$

Definition 2 *An* ***adjunction grammar*** *is a sign grammar where for each mode f:*

➊ *if $\Omega(f) = 0$ then f^ε is either a string or a context;*

➋ *if $\Omega(f) > 0$, f^ε is a $\Omega(f)$–ary polynomial formed with the help of constant strings, pair formation, and the operation $*$.*

(In fact, ➊ is just ➋ for the case $n = 0$.) Notice that the f^γ are the syntactic categories in the ordinary sense. Suppose by way of example that we have a single adjunction site in the following sentence, which we enclose by the pair of triangles $\triangledown \cdots \blacktriangle$ of category N:

(48) `The ▽ cat ▲ is under the table.`

Then this sentence is translated into the unary mode g, where ($C = \langle \vec{u}_1, \vec{u}_2 \rangle$)

(49) $$g^{\varepsilon}(C) = \langle \texttt{the}, \texttt{is under the table} \rangle * (C * \texttt{cat})$$
$$= \texttt{the}^\frown \vec{u}_1{}^\frown \texttt{cat}^\frown \vec{u}_2{}^\frown \texttt{is under the table}$$

(50) $$g^{\gamma}(P) = \begin{cases} \text{S} & \text{if } P = \text{N} \\ \text{undefined} & \text{else} \end{cases}$$

(For example, $\langle \vec{u}_1, \vec{u}_2 \rangle$ may be $\langle \texttt{red}, \varepsilon \rangle$ or $\langle \varepsilon, \texttt{which scratches me} \rangle$.) If we have two adjunction sites, say

(51) `The` ▽ `cat` ▲ `is under the` ▽ `table` ▲

Then we simply use a binary mode h ($C = \langle \vec{u}_1, \vec{u}_2 \rangle, D = \langle \vec{v}_1, \vec{v}_2 \rangle$):

(52) $$h^{\varepsilon}(C, D) = \langle \langle \texttt{the}, \varepsilon \rangle * (C * \texttt{cat}), \langle \texttt{under the}, \varepsilon \rangle * (D * \texttt{table}) \rangle * \texttt{is}$$
$$= \texttt{the}^\frown \vec{u}_1{}^\frown \texttt{cat}^\frown \vec{u}_2{}^\frown \texttt{is under the}^\frown \vec{v}_1{}^\frown \texttt{table}^\frown \vec{v}_2$$

(53) $$h^{\gamma}(P, Q) = \begin{cases} \text{S} & \text{if } P = Q = \text{N} \\ \text{undefined} & \text{else} \end{cases}$$

These examples should suffice for the reader to see that the above definition of adjunction grammar really captures the generative device of multiple parallel adjunction. It is however based on strings, not on trees.

The grammars so defined are a subclass of the Linear Context Free Rewriting Systems, in which constituents can have any number of parts (see [Kracht, 2003] for an exposition).

8 Conclusion

This paper vindicates the suspicion that adjunction is not compositional. We have shown that if one wants adjunction to be compositional, not only have derivations to be turned on their heads; also, we need to artificially split syntactic trees into several isomorphic copies to ensure that synonymy does not block our way. The result seems to carry over mutatis mutandis to the somewhat lesser known contextual grammars. It is true, though, that these grammars are much more flexible in the way adjunction can be restricted. They can, for example, define arbitrary context sets for the adjunction site, something which TAGs cannot do. Nevertheless, the present results show that there is no hope that a solution within contextual grammars is either simple or elegant.

The result does not disqualify adjunction as such, but it warns us against the assumption that what is natural as a syntactic operation will also be natural as a semantic operation. I guess it has always been clear in most people's intuition that adjunction just can't always be coroutined by semantics. [1]

References

[Frank, 2002] Robert Frank. *Phrase Structure Composition and Syntactic Dependencies.* MIT Press, Cambridge (Mass.), 2002.

[Harris, 1979] Zellig S. Harris. *Mathematical Structures of Language.* Robert E. Krieger Publishing Company, Huntington, New York, 1979.

[Hodges, 2001] Wilfrid Hodges. Formal features of compositionality. *Journal of Logic, Language and Information*, 10:7 – 28, 2001.

[Kallmeyer and Joshi, 2003] Laura Kallmeyer and Aravind Joshi. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. *Research in Language and Computation*, 1:3 – 58, 2003.

[Kamp and Reyle, 1993] Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Language and Discourse Representation.* Kluwer, Dordrecht, 1993.

[Kracht, 2003] Marcus Kracht. *The Mathematics of Language.* Mouton de Gruyter, Berlin, 2003.

[Marcus, 1967] Solomon Marcus. *Algebraic Linguistics; Analytical Models.* Academic Press, New York and London, 1967.

[Vermeulen, 1995] Kees F. M. Vermeulen. Merging without mystery or: Variables in dynamic semantics. *Journal of Philosophical Logic*, 24:405 – 450, 1995.

[1] Aravind Joshi (p.c.) has argued that meanings are assigned to derivations trees, not the trees that they yield. This claim is tantamount to giving up compositionality. For it implies that at any stage we have access to the history of the tree in addition to the tree itself.

[Zadrozny, 1994] Wlodek Zadrozny. From Compositional Semantics to Systematic Semantics. *Linguistics and Philosophy*, 17:329 – 342, 1994.