ADJUNCTION STRUCTURES AND SYNTACTIC DOMAINS

MARCUS KRACHT

Abstract. Since Chomsky's Barriers System, crucial use is made of the distinction between nodes and categories. In particular, Chomsky has shown that subjacency can be redefined in such a way that it looks like a tight command relation in the sense of Kracht [10] and yet allows for cyclical movement. However, this new shift has been accompanied by a great confusion concerning the structures about which we are now talking. In this paper we will propose a definition of *adjunction structures* that allows to encompass the distinction between nodes and categories. Moreover, we will adress many questions that ensue once these structures are defined.

1. Introduction

Ever since the introduction of the Barriers System in [3] categories are assumed to be complex objects, consisting of (possibly several) segments. Linguists have found many uses of the distinction between categories and their segments. The primary use of course was the distinction between domination of one category by another and inclusion of category by another. Unfortunately, it is not easy to get accustomed to this distinction. What is more, the effects of differentiating between categories and segments are usually looked at only superficially without awareness of the many awkward details. (A notable exception is [14].) However, it must be realized that the standard notation of structures as labelled trees is inappropriate when dealing with categories. The categories are not the nodes of the tree; nor is it appropriate to equate them with maximal connected sets of nodes with identical label (see Section 3.4). The categories are distinct objects introduced into the

This work has been carried out in the Innovationskolleg 'Formale Modelle kognitiver Komplexität' (INK 12) funded by the DFG. I wish to thank Carsten Grefe with whom I discussed many issues concerning adjunction structures. Some results have been worked out in collaboration with him. This work has been presented at workshops in Amsterdam, Tübingen and Prague. My thanks go to Patrick Blackburn, Tom Cornell, Hans–Martin Gärtner, Hans–Peter Kolb and Maarten de Rijke. Special thanks to Hap and Uwe for organizing a life–changing event in Prague.

tree structure that fundamentally change the structure of these (previously treelike) objects. To illustrate this (and to rid ourselves from this confusing term *category*) we have found it best to introduce the notion of a *block*. A block is a linear set of nodes. A node is a *segment* of a block if it is contained in it. We demand that each node is a segment of some block. If we equip trees with a partitioning of the node set into a set of blocks we get what we call *adjunction structures* (Section 3). These are structures in which we claim Post–Barriers linguistics should be done.

At first blush this seems to be just a harmless complication. However, many question arise that have never been answered in the literature (many of them have to our knowledge never even been raised). First of all, what is now a *constituent*? This must be determined for many reasons, among other in order to know what elements can undergo movement. Second, is there a way of defining structures in which categories are simple objects? In other words, can we discard nodes and define structures that contain categories in place of nodes? The answer is that we can, but we loose some information about the makeup of the adjunction structure (Section 4). It does seem, however, that for the purpose of linguistics the loss of information is inessential. Third, as regards the theory of command relations of [10] can it be adapted to adjunction structures such that the mathematical properties remain and we get the effects of the Barrier's type domains (that is, can we escape barriers by successive movement even though the relation looks like a tight relation)? The answer is yes and this will be shown in Section 5. Moreover, we will show in 5.3 that the command relations exhibit what we call *Movement Invariance*. This means that whether or not two nodes are in a certain relation at some level of derivation can be checked at the end of derivation. This is a welcome property, since it paves the way for stratification of transformational theory. (See Section 6.)

2. Trees

2.1. **General Structures.** By a natural number we understand a positive integer; 0 is a natural number. Moreover, the natural number n is also the set of its predecessors, that is, we put $n := \{0, 1, \ldots, n-1\}$. '$i < n$' is synonymous with '$i \in n$'. A **structure** of signature $\langle\langle \rho_i : i < m\rangle, \langle \sigma_i : i < n\rangle\rangle$ over a set M is a triple

$$\mathfrak{M} = \langle M, \langle R_i : i < m\rangle, \langle f_i : i < n\rangle\rangle$$

such that $R_i \subseteq M^{\rho_i}$ for each $i < m$ and $f_i : M^{\sigma_i} \to M$. Let $\mathfrak{N} = \langle N, \langle S_i : i < m\rangle, \langle g_i : i < n\rangle\rangle$ be a structure of same signature as $\mathfrak{M}$. A function $h : M \to N$ is an **isomorphism** from $\mathfrak{M}$ to $\mathfrak{N}$ if (i) h is bijective, (ii) for every $i < m$ and every sequence $\vec{x} = \langle x_j : j < \rho_i\rangle \in M^{\rho_i}$ we have $\vec{x} \in R_i$ iff $h(\vec{x}) \in S_i$, where $h(\vec{x}) := \langle h(\vec{x}_j) : j < \rho_i\rangle$, and (iii) for every $i < n$ and every sequence $\vec{x} \in M^{\sigma_i}$, $h(f_i(\vec{x})) = g_i(h(\vec{x}))$. Given a structure $\mathfrak{M}$ over a set M, and a bijection $h : M \to N$ then there exists exactly one structure $\mathfrak{N}$ over N which makes h an isomorphism. We say that h **induces** $\mathfrak{N}$ over N.

2.2. **Tree Structures.** Let S be a set and $<$ a binary relation on S. $<$ is called **transitive** if $x < y$ and $y < z$ imply $x < z$, **irreflexive** if for no $x \in S$, $x < x$; $<$ is **linear** if for $x, y \in S$ either $x < y$, $x = y$ or $y < x$. A **linear order** is a pair $\langle S, <\rangle$ such that $<$ is transitive, irreflexive and linear. If $\mathfrak{S} = \langle S, <\rangle$ is a finite linear order and S has n elements, $\mathfrak{S}$ is isomorphic to the order $\langle n, <\rangle$, where by our convention, $n = \{0, 1, \ldots, n-1\}$ and $<$ is the usual order on the natural numbers. Given a nonempty set D a triple $\vec{x} = \langle S, <, \ell\rangle$ is called a D–**string** if $\langle S, <\rangle$ is a linear order and $\ell : S \to D$. If S has n members, $\vec{x}$ is said to be of **length** n. Alternatively, a string of length n can be thought of as a function $f : \{0, 1, \ldots, n-1\} \to D$. A string can also be viewed as a list, namely the list $\langle f(i) : i < n\rangle$. A **domination structure** is a pair $\mathfrak{T} = \langle T, <\rangle$ where T is a nonempty set and $<$ a binary relation on T which is irreflexive and transitive. We write $x \leq y$ if $x < y$ or $x = y$. x and y are **comparable** if $x \leq y$ or $y \leq x$. If $x < y$ we say that x is **properly dominated** by y. If $x < y$ and for no z we have $x < z < y$ then we say that y **immediately dominates** (**is a mother of**) x and write $y \succ x$ or $x \prec y$. If $x \prec y$, $z \prec y$ and $x \neq z$ we say that x and z are **sisters**. An element x such that $x \geq y$ implies $y = x$ is called a **leaf** or a **terminal node**. A **leaf of** x is a leaf u such that $u \leq x$. A node x is **preterminal** if for all y immediately dominated by x, y is a leaf. x **branches** if x has at least two daughters. We write

$$\begin{array}{rcl} \uparrow x & := & \{y : y \geq x\} \\ \downarrow x & := & \{y : y \leq x\} \end{array}$$

and call $\uparrow x$ the **upper cone** of x and $\downarrow x$ the **lower cone** of x. A **forest** is a domination structure in which all upper cones are linear (with respect to $<$). A triple $\mathfrak{T} = \langle T, r, <\rangle$ is a **tree** if $\langle T, <\rangle$ is a forest and $r \in T$ is such that $r > x$ for all $x \neq r$. r is called the **root** of $\mathfrak{T}$. Given a subset $U \subseteq T$, $<$ induces a binary relation $<_U := \{\langle x, y\rangle : x < y, x \in U, y \in U\}$. We write $<$ for $<_U$. We call $\wp(x) := \langle \uparrow x - \{x\}, r, <\rangle$ the **position** of x, and $c(X) := \langle \downarrow x, x, <\rangle$ the

constituent of x. It is clear that these structures are trees. It is to be noted that while the constituent headed by x contains x, the position of x does not. Often we will also call a set of the form $\downarrow x$ a constituent.

Let $\langle T, r, <\rangle$ be a tree. A binary relation $L \subseteq T \times T$ is **a (linear) ordering compatible with** $<$ if (ℓ) L is an irreflexive linear ordering on the leaves, and (c) $x\, L\, y$ iff $u\, L\, v$ for all leaves u, v such that $u \leq x$ and $v \leq y$. By (c), if $x \geq x'$, $y \geq y'$ and $x\, L\, y$ then $x'\, L\, y'$. Conversely, if x is not a leaf then $x\, L\, y$ iff for all $x' < x$, $x'\, L\, y$; likewise for y. Given L, we say that x **precedes** y if $x\, L\, y$ and for no z we have $x\, L\, z\, L\, y$ we say that x **immediately precedes** y. If x_1 and x_2 are sisters and $x_1\, L\, x_2$, then x_1 is called a **left sister of** x_2. If in addition x_1 immediately precedes x_2 then x_1 is called an **immediate left sister of** x_2. An **ordered tree** is a quadruple $\langle T, r, <, L\rangle$ where $\langle T, r, <\rangle$ is a tree and L an ordering relation on T compatible with $<$.

In general, there exists more than one ordering relation compatible with $<$. There is however one which is unique in the following sense. Call x and y **overlapping** and write $x \circ y$ if neither $x\, L\, y$ nor $y\, L\, x$. In general, comparable nodes overlap. For if $x \leq y$ then every leaf of x is a leaf of y. Moreover, there is at least one leaf $z \leq y$ such that $z \leq x$, and this shows that y cannot precede x either. Call $\langle T, r, <, L\rangle$ **exhaustively ordered** and L **exhaustive for** $\langle T, r, <\rangle$ if x and y overlap iff they are incomparable. It can be shown that there always exists an exhaustive ordering. Namely, for each x let $\delta(x)$ be the set of daughters of x. (If x is a leaf then $\delta(x) = \varnothing$.) Let $P_x \subseteq \delta(x) \times \delta(x)$ be a linear ordering of the daughters of x. Put $P := \bigcup_{x\in T} P_x$. Now put $u\, L(P)\, v$ if there exists z and daughters x, y of z such that $y \neq x$, $u \leq x$, $v \leq y$ and xPy. This ordering is exhaustive. For let u and v be incomparable. Then there exist a unique z and unique daughters x and y of z such that $y \neq x$, $u \leq x$ and $v \leq y$. P is linear on the daughters. Therefore $x\, P\, y$ or $y\, P\, x$. By definition of $L(P)$, in the first case $u\, L(P)\, v$, and in the second case $v\, L(P)\, u$. Only one of the two obtains. Now let x and y be comparable. Then first of all they are not daughters of the same node, so they are not comparable via P. Second, there exists no z and distinct daughters y, x such that $u \leq x$ and $v \leq y$. Hence, neither $u\, L(P)\, v$ nor $v\, L(P)\, u$ obtains.

2.3. **Labelled Structures.** All structures can also be equipped with a labelling. Let D be a set. A D–**structure** (of signature σ) is a a tuple of the form $\mathfrak{M}^+ := \langle M, \langle R_i : i < m\rangle, \langle f_i : i < n\rangle, \ell\rangle$ where $\mathfrak{M} = \langle M, \langle R_i : i < m\rangle, \langle f_i : i < n\rangle\rangle$ is a structure of signature σ and $\ell : M \to D$ a function, called the **labelling function**. $\ell(x)$ is called the **label** of x. A map $h : M \to N$ of sets is a homomorphism of D–structures if it is a homomorphism of the underlying structures, and x and $h(x)$ receive the same label. For example, a **labelled tree** (ordered or unordered) with labels in D is a tree together with a function $\ell : T \to D$, D the set of labels. Seen this way a D–string is a labelled linear order. Now let $\mathfrak{T} = \langle T, r, <, L, \ell\rangle$ be an ordered labelled tree with labels in D. Let $M \subseteq T$ be a set which is linearly ordered by L and such that for no $M \subsetneq N \subseteq T$, N is linearly ordered by L. Such sets are called **cuts**. If M is a cut, $\langle M, L, \ell\rangle$ is a D–string. This string is called the **string cut** of $\mathfrak{T}$ based on M. If M is the set of terminal nodes, M is a cut. The string cut based on M is called the **string associated with** $\mathfrak{T}$.

2.4. **Domains.** Barker and Pullum [2] observed that many relations in syntax can be defined in a uniform way. They defined the notion of a *command relation*. This has been taken over in Kracht [10]. We shall follow the outline of [10]. First, let $R \subseteq T^2$ be a binary relation on T and let $x \in T$. Put $xR := \{y : xRy\}$ and call xR the R–**domain** of x. If $y \in xR$, x is said to R–**command** y. A relation on a tree $\langle T, r, <\rangle$ is a **command relation** if

(1) The domain of r is the entire tree.
(2) The domain of a node $\neq r$ is a constituent properly containing x.
(3) If $x \leq y$ then $xR \subseteq yR$.

The last property is called *monotonicity*. A binary relation on an ordered labelled tree $\langle T, r, <, L, \ell\rangle$ is a command relation if it is a command relation on $\langle T, r, <\rangle$. Given a command relation R there is a function $f_R : T \to T$ such that $\downarrow f_R(x) = xR$ for all $x \in T$. This is called the **generating function** of R. The above conditions boil down to the requirements

(1) $f_R(r) = r$.
(2) If $x < r$ then $f_R(x) > x$.
(3) If $x \leq y$ then $f_R(x) \leq f_R(y)$.

A command relation is **tight** if it satisfies

Tightness. If $y < f_R(z)$ then $f_R(y) \leq f_R(z)$.

As Kracht [10] emphasizes, the command relations used in syntactic theory can be generated from very simple relations using intersection and relation composition.

Definition 1. Let D be a set of labels and $O \subseteq D$. Let $\langle T, r, <, L, \ell \rangle$ be an ordered D–tree. Then

$$\kappa(O, \mathfrak{T}) := \{\langle x, y \rangle : (\forall z > x)(\ell(z) \in O \to z \geq y)\} .$$

If $\langle x, y \rangle \in \kappa(O, \mathfrak{T})$ we say that x **O–commands** y. We often write O in place of $\kappa(O, \mathfrak{T})$.

Proposition 2. *The generating function for O–command is the function g_O mapping x onto the least node $y > x$ such that $\ell(y) \in O$, if it exists, and onto r otherwise. The relation O is tight for all $O \subseteq D$.*

Proof. With $R := \mathsf{O}$ let $g_O := f_R$. We show that g_O has the properties announced in the theorem. Let $y < g_O(x)$. If $g_O(x) = r$ the claim is certainly true. Let u be the least node $> y$ such that $\ell(u) \in O$. Such a node exists, since $g_O(x) < r$ and so $\ell(g_O(x)) \in O$. Moreover $y < g_O(x)$. Hence, $u \leq g_O(x)$, and $u = g_O(y)$. □

We have $\kappa(O, \mathfrak{T}) \supseteq \kappa(P, \mathfrak{T})$ if $O \subseteq P$ and $\kappa(O \cup P, \mathfrak{T}) = \kappa(O, \mathfrak{T}) \cap \kappa(P, \mathfrak{T})$. We call a tight relation **definable** if it is of the form O for a certain O. Now given two binary relations R, S on T define

$$R \circ S := \{\langle x, z \rangle : (\exists y \in T)(x\ R\ y\ R\ z)\}$$

We note here that

$$f_{R \circ S} = f_S \circ f_R$$

For if $f_{R \circ S}(x) = z$, then $z \geq x$ and there exists a y such that $x \leq y \leq z$ and $xRySz$. By monotonicity of the relations we may assume $y = f_R(x)$. Then $z = f_S(y) = f_S(f_R(x))$.

Definition 3. A binary relation R on a labelled (ordered) tree is a **chain** if there exist n and $C_i \subseteq D$, $i < n$, so that $R = \mathsf{C}_0 \circ \mathsf{C}_1 \circ \ldots \circ \mathsf{C}_{n-1}$, with $\mathsf{C}_i := \kappa(C_i, \mathfrak{T})$. R is called **definable** if it is an intersection of finitely many chains.

Theorem 4. *The set of definable command relations is closed under intersection, union and relation composition.*

The proof of this theorem can be found in Kracht [10]. As a corollary we note that there is a smallest and a largest command relation. Both are tight. The smallest relation is obtained when we choose $O := D$, the entire set of labels. Then, x O–commands y iff $x = r$ or the node immediately dominating x dominates y. This relation is called **idc–command**. The largest relation is obtained when $O := \varnothing$. Here, all nodes O–command all other nodes. (This relation is only of mathematical importance.)

2.5. **The Use of Domains.** In Kracht [10] it was claimed that relations in GB theory are definable command relations. It was shown that the system of Koster [9] can be reformulated using definable command relations. Here we will give some more examples, this time of some more canonical literature. First, the notion of *c–command* is of central importance. It is usually defined in two ways, depending on the authors. It is either identical to *idc–command* or identical to *max–command*. Here, the relation of *max–command* is obtained by choosing O to be the set of maximal (= phrasal) nodes. Indeed, with this choice given, x max–commands y if all phrasal nodes properly dominating y also dominate y. In many cases of the literature it is also required that x and y are incomparable. We call this the *Non Overlapping Condition*. It has been argued in [2] that to include this condition into the definition of a command relation is not a good choice. [1] Suffice it to say that from a mathematial point of view it is better to do without the *Non Overlapping Condition*.

For the purposes of binding theory and other modules such as Case–Theory the relation of c–command (which we now take to be *idc–command*) is central. Indeed, it is also the smallest command relation, and used quite extensively as a diagnostic instrument to analyse the D–Structure of sentences, using evidence from binding theory (for example, see Haider [7]). In Baker [1] it has been modified somewhat, but this modification will be automatically implemented in adjunction structures.

Let us now turn to some more difficult questions, namely the nonlocal relations. Here, the most prominent one is *subjacency*. In its most primitive form it says that a constituent may not move across more than one bounding node. This condition can be rephrased easily in the present framework. Let BD be the set of labels corresponding to bounding nodes. Examples are $BD = \{S, NP\}$ or $BD = \{S', NP\}$, in the LGB–terminology. The choice between these sets is empirical and does not touch on the question how to define *subjacency*. Now, the requirement on subjacency can be rephrased as a condition on the relation between the trace and the antecedent. Let BD be the relation of BD–command. Then put

$$\mathsf{SUB} := \mathsf{BD} \circ \mathsf{BD}$$

We claim that $y \in x\,\mathsf{SUB}$ — that is, x SUB–commands y — iff y is subjacent to x. For to check this, we need to look at the least node z such that z dominates both x and y. Suppose that x is subjacent

[1] This condition can also be formulated in many other ways. One is to require that x precedes y or y precedes x. We will not discuss that issue here.

to y. Then in the set $[x, z] - \{x\}$ at most two nodes carry a label from *BD*. Let f be the generating function of *BD*–command. Then the generating function of subjacency is $f \circ f$. By definition, $f(x)$ is either a node carrying a label in *BD*, or $f(x) = r$. Hence, for the node z defined above, $z \leq f \circ f(x)$. It follows that x *SUB*–commands y. Now let conversely x *SUB*–command y. Let again z be the least node dominating both x and y. Then $z \leq f \circ f(x)$ and it is easy to see that at most two nodes of label $\in$ *BD* can be in the set $[x, z] - \{x\}$. So, y is subjacent to x.

In Chomsky's Barriers System (see [3]) this definition of movement domain has been attacked on the ground that it is empirically inadequate. The definition that Chomsky gives for subjacency makes use of adjunction structures and an appropriate adaptation of the definition of command relations for those structures. We will return to that question. However, the use of adjunction is not necessary to get at most of the facts that the new notions and definitions are intended to capture. It would take too much time to prove this claim. We will be content here with outlining how different notions of movement domains can achieve the same effect. Crucially, the instrument we are using is that of composing relations. As in the definition of subjacency given above, the relation is defined from tight command relations by means of relation composition. This is no accident. It can be shown that tight command domains cannot be escaped by movement (whence the name). However, subjacency is intended to be a one–step nearness constraint, and it can clearly be violated in a successive step. Now consider the following sentence

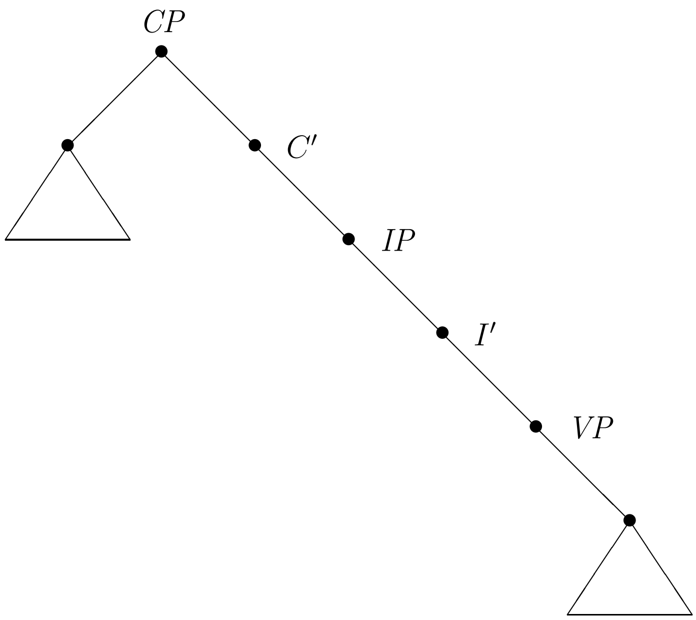
(1.) *[Von welcher Stadt]$_1$ hast Du [den Begin [der Zerstörung t_1]] gesehen?*
[Of which city]$_1$ did you witness [the beginning of [the destruction t_1]]?

Here, in moving the *wh*–phrase (pied–piping the preposition), two bounding nodes have been crossed. Indeed, to capture *wh*–movement it seems more plausible not to count intervening nominal heads. If that is so, let us look for an alternative. It has been often suggested that the only escape hatch for a *wh*–phrase is the specifier of *comp*. A *wh*–phrase always targets the next available spec–of–comp. If that spec is filled, movement is blocked. To implement this we take advantage of the fact that the complement of C^0 is *IP*. Hence we propose the following domain

$$\mathsf{WHM} := \mathsf{IP} \circ \mathsf{CP}$$

Fig. 1 below illustrates a case of wh–movement, where a wh–phrase is movemed from inside a verb phrase into spec–of–comp. The domain of the trace is the least *CP* above the least *IP* which is above the

Figure 1. Wh–Movement



trace. In the present case, the domain is the clause containing that trace. From spec–of–comp, however, the domain would be the next higher clause! This readily accounts for the fact that in a subsequent movement step the wh–phrase may target the next higher spec–of–comp. (Of course, the present domain allows the constituent to move anywhere within that domain. We assume however that independent conditions will ensure that only this position is chosen, if at all it is available.) Although this definition may have its problems, too, what we have shown is that ideas proposed in the literature about movement can be succinctly rephrased using definable domains.

3. Adjunction Structures

3.1. **Adjunction Structures.** An **adjunction structure** is a structure $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ where $\langle S, r, <\rangle$ is a tree and $\mathfrak{C}$ a partitioning of S into subsets which are linear with respect to $<$. (A note. To make this a structure in the sense above, we would have to use instead of $\mathfrak{C}$ the relation $\sim$ defined by $x \sim y$ iff there exists a $\mathfrak{x} \in \mathfrak{C}$ such that $x \in \mathfrak{x}$ and $y \in \mathfrak{x}$. Conversely, given the equivalence relation $\sim$, we put $\mathfrak{C} := \{[x]_\sim : x \in S\}$, where $[x]_\sim := \{y : y \sim x\}$. It is however in many instances easier to work with $\mathfrak{C}$ rather than $\sim$.) We say that

$\mathfrak{S}$ is **based** on S. A member of $\mathfrak{C}$ is called a **block**. Thus a block is always of the form $[x, y] = \{z | x \leq z \leq y\}$ for some $x, y \in S$. Members of a block are called **segments** of that block. We let $\mathfrak{b}_\circ$ be the minimal segment of $\mathfrak{b}$ and $\mathfrak{b}^\circ$ the maximal segment of $\mathfrak{b}$. By definition, $\mathfrak{b} = [\mathfrak{b}_\circ, \mathfrak{b}^\circ]$. Given x, there is exactly one $\mathfrak{x} \in \mathfrak{C}$ such that $x \in \mathfrak{x}$; we denote $\mathfrak{x}$ by $b(x)$. Thus, $b : S \to \mathfrak{C}$ is a function assigning to a node the block in which that node sits. In sequel many notions for trees will be defined for adjunction structures. Generally, blocks will take over the part of nodes. Given $\mathfrak{S} = \langle S, <, r, \mathfrak{C} \rangle$ and a subset $U \subseteq S$, put as before $<_U := \{\langle x, y \rangle : x \in U, y \in U, x < y\}$. Suppose that there is a $u \in U$ such that $x \leq u$ for all $x \in U$. Then $\mathfrak{S} \upharpoonright U$ defined below is an adjunction structure.

$$\mathfrak{S} \upharpoonright U := \langle U, <_U, u, \{\mathfrak{x} \cap U : \mathfrak{x} \in \mathfrak{C}, \mathfrak{x} \cap U \neq \varnothing\} \rangle$$

We say $\mathfrak{S} \upharpoonright U$ is the adjunction structure **induced by $\mathfrak{S}$ on** U. $\mathfrak{S} \upharpoonright U$ is called a **subadjunction structure** if U is a union of blocks and lower closed. (It follows that $U = \downarrow \mathfrak{x}^\circ$ for some $\mathfrak{x}$.)

An ordered adjunction structure is based on an ordered tree. An adjunction structure over D is an adjunction structure together with a function $\ell : \mathfrak{C} \to D$. Equivalently, we may say that for the corresponding node based function $\ell \circ b$ that is a function from S to D such that nodes of the same block receive identical label. Given two blocks, $\mathfrak{b}$ and $\mathfrak{c}$, there are two notions of superiority, called *containment* and *inclusion*. $\mathfrak{b}$ **contains** $\mathfrak{c}$ if some segment of $\mathfrak{b}$ properly dominates some (and therefore all) segments of $\mathfrak{c}$. We write $\mathfrak{b} > \mathfrak{c}$ (and $\mathfrak{b} \geq \mathfrak{c}$ if $\mathfrak{b} = \mathfrak{c}$ or $\mathfrak{b} > \mathfrak{c}$). $\mathfrak{b}$ **includes** $\mathfrak{c}$, in symbols $\mathfrak{b} \gg \mathfrak{c}$, if all segments of $\mathfrak{b}$ properly dominate all segments of $\mathfrak{c}$. If $\mathfrak{b}$ includes $\mathfrak{c}$, it also contains $\mathfrak{c}$. The following characterization of containment and inclusion can be given.

Proposition 5. (1) $\mathfrak{b} > \mathfrak{c}$ *iff* $\mathfrak{b}^\circ > \mathfrak{c}^\circ$.
(2) $\mathfrak{b} \geq \mathfrak{c}$ *iff* $\mathfrak{b}^\circ \geq \mathfrak{c}^\circ$ *iff* $\mathfrak{b}^\circ \geq \mathfrak{c}_\circ$.
(3) $\mathfrak{b} \gg \mathfrak{c}$ *iff* $\mathfrak{b}_\circ > \mathfrak{c}^\circ$ *iff* $\mathfrak{b}_\circ > \mathfrak{c}_\circ$.

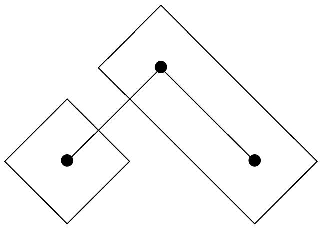
We note the following.

Corollary 6. *$<$ and $\ll$ are irreflexive and transitive.*

Proof. Since $\mathfrak{b}^\circ < \mathfrak{b}^\circ$ cannot hold, $<$ is irreflexive, by (1.) of the previous theorem. Furthermore, let $\mathfrak{b} > \mathfrak{c} > \mathfrak{d}$. Then $\mathfrak{b}^\circ > \mathfrak{c}^\circ > \mathfrak{d}^\circ$. Hence, by transitivity, $\mathfrak{b}^\circ > \mathfrak{d}^\circ$, from which $\mathfrak{b} > \mathfrak{d}$. Now we turn to $\ll$. Since $\mathfrak{b}_\circ > \mathfrak{b}_\circ$ cannot hold, $\ll$ is irreflexive. Now let $\mathfrak{b} \gg \mathfrak{c} \gg \mathfrak{d}$. Then $\mathfrak{b}_\circ > \mathfrak{c}_\circ > \mathfrak{d}_\circ$. By transitivity, $\mathfrak{b}_\circ > \mathfrak{d}_\circ$. Hence $\mathfrak{b} \gg \mathfrak{d}$. □

It can also be shown that (1.) if $\mathfrak{a} < \mathfrak{b} \ll \mathfrak{c}$ then $\mathfrak{a} \ll \mathfrak{c}$ and (2) if $\mathfrak{a} < \mathfrak{b} < \mathfrak{c}$ and $\mathfrak{a} \ll \mathfrak{c}$ then $\mathfrak{b} \ll \mathfrak{c}$. As we will see later, this

Figure 2. A Complex Morphological Head



completely characterizes the properties of $<$ and $\ll$ with respect to the block structure. We define

$$\begin{array}{lcl} \uparrow \mathfrak{b} & := & \{\mathfrak{c} : \mathfrak{c} \geq \mathfrak{b}\} \\ \downarrow \mathfrak{b} & := & \{\mathfrak{c} : \mathfrak{c} \leq \mathfrak{b}\} \\ \twoheaduparrow \mathfrak{b} & := & \{\mathfrak{b}\} \cup \{\mathfrak{c} : \mathfrak{c} \gg \mathfrak{b}\} \\ \twoheaddownarrow \mathfrak{b} & := & \{\mathfrak{b}\} \cup \{\mathfrak{c} : \mathfrak{c} \ll \mathfrak{b}\} \end{array}$$

and call the adjunction structures based on the sets $\uparrow \mathfrak{b} - \{\mathfrak{b}\}$ and $\twoheaduparrow \mathfrak{b} - \{\mathfrak{b}\}$ the **weak** (**strong**) **position** and $\downarrow \mathfrak{b}$ and $\twoheaddownarrow \mathfrak{b}$ the **weak** (**strong**) **constituent** of $\mathfrak{b}$.

Proposition 7. *Let $\langle S, r, <, \mathfrak{C}\rangle$ be an adjunction structure. Let $\mathfrak{r} \in \mathfrak{C}$ be the block containing r. Then $\langle \mathfrak{C}, \mathfrak{r}, <\rangle$ is a tree and $\langle \mathfrak{C}, \ll\rangle$ is a forest.*

Proof. Clearly, the block containing r contains all blocks. Therefore, we only have to show that weak and strong upper cones are linear. First, weak upper cones. So, let $\mathfrak{c} > \mathfrak{b}$ and $\mathfrak{d} > \mathfrak{b}$. Then $\mathfrak{c}^\circ > \mathfrak{b}^\circ$ as well as $\mathfrak{d}^\circ > \mathfrak{b}^\circ$. So, $\mathfrak{c}^\circ$ and $\mathfrak{d}^\circ$ are comparable. Then either $\mathfrak{c}^\circ < \mathfrak{d}^\circ$, $\mathfrak{c}^\circ = \mathfrak{d}^\circ$ or $\mathfrak{c}^\circ > \mathfrak{d}^\circ$. In the first case, $\mathfrak{c}$ is contained in $\mathfrak{d}$, in the second case they are equal, and in the third case case $\mathfrak{c}$ contains $\mathfrak{d}$. Now for the strong upper cones. Let $\mathfrak{c} \gg \mathfrak{b}$ and $\mathfrak{d} \gg \mathfrak{b}$. Then $\mathfrak{c}_\circ > \mathfrak{b}^\circ$ as well as $\mathfrak{d}_\circ > \mathfrak{b}^\circ$. Therefore, $\mathfrak{c}^\circ$ and $\mathfrak{d}^\circ$ are comparable. If they are equal $\mathfrak{c} = \mathfrak{d}$. Otherwise, let $\mathfrak{c}_\circ > \mathfrak{d}_\circ$. Then, by linearity of the blocks, $\mathfrak{c}_\circ > \mathfrak{d}^\circ$, so that $\mathfrak{d}$ is included in $\mathfrak{c}$. Similarly if $\mathfrak{d}_\circ > \mathfrak{c}_\circ$. $\Box$

The adjunction structure below illustrates that $\ll$ does not need to be a tree ordering. Here $\ll = \varnothing$. We will see below that this is important. It allows for complex heads in syntax. Namely, we propose the following definition.

Definition 8. Let $\mathfrak{A}$ be an adjunction structure. A **morphological head** of $\mathfrak{A}$ is a subadjunction structure of $\mathfrak{A}$ in which $\ll = \varnothing$.

This conforms to the standard use of head–adjunction. Heads adjoin to heads to form complex heads, and they act in syntax as words (though they receive a structural analysis). The rationale of the definition is the following. If there is a pair $\mathfrak{x}$ and $\mathfrak{y}$ of blocks such that $\mathfrak{x}$ is immediately included in $\mathfrak{y}$, then there exists a pair $\mathfrak{x}$ and $\mathfrak{y}$ such that $\mathfrak{x}$ does not include any block. Then $\mathfrak{x}$ is a zero level projection. If $\mathfrak{x}$ is the (relational) head of $\mathfrak{y}$, $\mathfrak{y}$ is a projection of $\mathfrak{x}$, and so the constituent contains a nonminimal projection. If $\mathfrak{x}$ is not the head, then we look at the sisters of $\mathfrak{x}$. Some sister, call it $\mathfrak{z}$, is the head of $\mathfrak{y}$, and either it is minimal (then we are done) or it is not. In the latter case the constituent headed by $\mathfrak{z}$ contains a pair $\mathfrak{t}$ and $\mathfrak{u}$ such that $\mathfrak{t} \ll \mathfrak{u}$. Since the constituent $\mathfrak{z}$ is smaller than the original one, this procedure will come to an end and we get a pair of blocks such that one is a nonminimal projection of the other. Hence what we have defined is exactly what is standard. In order not to get confused with the usual relational notion of a head in a construction we call them *morphological* heads.

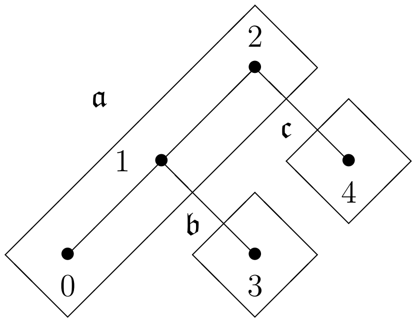
3.2. **Ordering on Adjunction Structures.** Given an ordering L compatible with $<$ put $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$ iff for some $x \in \mathfrak{b}$ and $y \in \mathfrak{c}$, $x\,L\,y$. It turns out that $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$ iff $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ$. For if $x \in \mathfrak{b}$ and $y \in \mathfrak{c}$, then $\mathfrak{b}_\circ \leq x$ and $\mathfrak{c}_\circ \leq y$. If $x\,L\,y$, then $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ$. Conversely, assume that $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ$. Then $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$ by definition. The problem of compatibility of this order with the dominance structures arises. This can only be solved if we define the notion of a *leaf* first.

Definition 9. A **leaf** in an adjunction structure is a block which contains a leaf of the tree. Equivalently, a block $\mathfrak{a}$ is a leaf iff $\mathfrak{a}$ is $\ll$–minimal.

Proposition 10. $\underline{L}$ *is an ordering on* $\mathfrak{C}$ *compatible with* $\ll$.

Proof. Let $\mathfrak{S}$ be an adjunction structure based on S. *$\underline{L}$ is irreflexive and transitive.* If $\mathfrak{b}\,\underline{L}\,\mathfrak{b}$ then $\mathfrak{b}_\circ\,L\,\mathfrak{b}_\circ$; which is not the case. Now suppose that $\mathfrak{b}\,\underline{L}\,\mathfrak{c}\,\underline{L}\,\mathfrak{d}$. Then $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ\,L\,\mathfrak{d}_\circ$. By transitivity of L, $\mathfrak{b}_\circ\,L\,\mathfrak{d}_\circ$, and so $\mathfrak{b}\,\underline{L}\,\mathfrak{d}$. *$\underline{L}$ satisfies (ℓ).* Let $\mathfrak{b}$ and $\mathfrak{c}$ be minimal with respect to $\ll$. Then $\mathfrak{b}_\circ$ and $\mathfrak{c}_\circ$ are leaves. Thus, $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ$, $\mathfrak{b}_\circ = \mathfrak{c}_\circ$ or $\mathfrak{c}_\circ\,L\,\mathfrak{b}_\circ$. In the first case we have $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$ and in the third case $\mathfrak{c}\,\underline{L}\,\mathfrak{b}$. In the second case we have $\mathfrak{b} = \mathfrak{c}$, since the blocks partition S. *$\underline{L}$ satisfies (c).* Assume that for all $\ll$–minimal $\mathfrak{f} \ll \mathfrak{b}$ and all $\ll$–minimal $\mathfrak{g} \ll \mathfrak{c}$, $\mathfrak{f}\,\underline{L}\,\mathfrak{g}$. Then for all such $\mathfrak{f}$ and $\mathfrak{g}$, $\mathfrak{f}_\circ\,L\,\mathfrak{g}_\circ$ and $\mathfrak{f}_\circ < \mathfrak{b}_\circ$, $\mathfrak{g}_\circ < \mathfrak{c}_\circ$. By compatibility, $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ$, from which $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$, by definition. Conversely, let $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$ and let $\mathfrak{f}$ be a $\ll$–minimal element in $\downarrow\mathfrak{b}$, $\mathfrak{g}$ a $\ll$–minimal element in $\downarrow\mathfrak{c}$. Then $\mathfrak{f}_\circ \leq \mathfrak{b}_\circ$ and $\mathfrak{g}_\circ \leq \mathfrak{c}_\circ$. Now, $\mathfrak{b}_\circ\,L\,\mathfrak{c}_\circ$ since $\mathfrak{b}\,\underline{L}\,\mathfrak{c}$, and so $\mathfrak{f}_\circ\,L\,\mathfrak{g}_\circ$, by compatibility. Hence $\mathfrak{f}\,\underline{L}\,\mathfrak{g}$. ∎

FIGURE 3



$\underline{L}$ is not necessarily compatible with $<$. Let

$$\begin{array}{lcl} S & := & \{0,1,2,3,4\}, \\ < & := & \{\langle 0,1\rangle, \langle 0,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 4,2\rangle\}, \\ \underline{L} & := & \{\langle 0,3\rangle, \langle 0,4\rangle, \langle 3,4\rangle\}, \\ \mathfrak{C} & := & \{\{0,1,2\}, \{3\}, \{4\}\}. \end{array}$$

Put $\mathfrak{a} := \{0,1,2\}$, $\mathfrak{b} := \{3\}$ and $\mathfrak{c} := \{4\}$. Then $\mathfrak{a} > \mathfrak{b}$ and $\mathfrak{a} > \mathfrak{c}$. Moreover, since $0 = \mathfrak{a}_\circ$, and $0\,L\,3$ we have $\mathfrak{a}\,\underline{L}\,\mathfrak{b}$. So, $\mathfrak{a}$ is comparable with a node that it dominates. This is excluded. (For otherwise by (c), $\mathfrak{b}\,L\,\mathfrak{b}$, a contradiction.)

One may ask whether by a different definition of $\underline{L}$ we can make it compatible with both $<$ and $\ll$. We may instead of the lower segments compare the upper segments. This however does not yield satisfactory results. First, the reader may check that if $\mathfrak{a}\,\underline{L}\,\mathfrak{b}$ then either $\mathfrak{a}_\circ\,L\,\mathfrak{b}^\circ$ or $\mathfrak{a}^\circ\,L\,\mathfrak{b}_\circ$. However, it is not predictable which of the two obtains. Hence, the remaining notion would be $\mathfrak{a}\,\widehat{L}\,\mathfrak{b}$ iff $\mathfrak{a}^\circ\,L\,\mathfrak{b}^\circ$. This ordering fails to be linear on the leaves, as the above example shows. Instead, putting a very mild requirement on the block structure is enough to ensure that L is compatible with $<$ as well.

Definition 11. An adjunction structure $\mathfrak{S}$ is **proper** if a block is $<$–minimal exactly when it is $\ll$–minimal. $\mathfrak{S}$ is called **standard** if (1) nodes are branching iff they are neither terminal nor preterminal, (2) for all terminal nodes x, $\{x\}$ is a block. $\mathfrak{S}$ is **2–standard** if it is standard and nodes have at most two daughters.

It is not hard to see that standard structures are proper. Grammars in Chomsky Normal form generate 2–standard structures (assuming *Maximal Blocks*, defined below).

Proposition 12. *Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be proper and L compatible with the tree ordering. Then let $\underline{L} \subseteq \mathfrak{C}^2$ be defined by $\mathfrak{x}\,\underline{L}\,\mathfrak{y}$ iff for all leaves*

$\mathfrak{a}$ *and* $\mathfrak{b}$ *such that* $\mathfrak{a} \leq \mathfrak{x}$ *and* $\mathfrak{b} \leq \mathfrak{y}$ *we have* $\mathfrak{a} \,\underline{L}\, \mathfrak{b}$*.* $\underline{L}$ *is compatible with both* $\ll$ *and* $<$*.*

Definition 13. An **ordered adjunction structure** is a quintuple

$$\langle \mathfrak{S}, r, <, \underline{L}, \mathfrak{C} \rangle$$

such that $\langle \mathfrak{S}, r, <, \mathfrak{c} \rangle$ is an adjunction structure and $\underline{L}$ an ordering compatible with $\ll$.

3.3. **What Is a Constituent?** Clearly, we must answer the question of what is a *constituent* in an adjunction structure $\mathfrak{S}$. Of course, constituents of $\mathfrak{S}$ must be adjunction structures, and we demand that they be of the form $\mathfrak{S} \upharpoonright U$ for some U. It follows immediately that $U = \downarrow x$ for some x. We have two choices: we may either demand x to be a lower segment of some block, or an upper segment of some block. In the first case we say that a constituent must be based uniformly on a *strong cone* or on a *weak cone*. If we opt for weak cones, then the notion of 'constituent of $\mathfrak{S}$' and 'subadjunction structure of $\mathfrak{S}$' coincide. We will use movement as a diagnostic instrument for determining what nature the constituents have. That we can do so is due to the fact that there exist weak cones which are not strong and strong cones which are not weak. If we find such cones being moved, this gives decisive evidence for our choice. Such evidence comes from *head–movement*. The current analysis is that head–movement uniformly creates complex morphological heads. So, if V moves to I it creates the complex head $[V\ I]$, and if that head moves to C it creates the complex head $[[V\ I]\ C]$. Now, if strong cones may move it may be possible for the I in $[V\ I]$ to move alone and create the complex head $[C\ I]$. As we say, I *excorporates* from $[V\ I]$. Generally, excorporation is not an option. (Notice that even if constituents are weak, there is an option for V to excorporate. That must be ruled out independently.) What is more, the weak cone $[V\ I]$ is not a strong cone. In fact, no complex head is a strong cone; for $\ll$ is always empty. Hence complex heads are never of the form $\downarrow \mathfrak{x}$ for an $\mathfrak{x}$. Hence, with respect to heads, only weak cones may move. Although it is not necessary to assume that phrasal movement moves weak cones, it would be quite unsatisfactory if we did. For then the uniformity of the movement operation is seriously challenged. There are also other reasons to assume that constituents are weak cones. One is that for weak cones it also holds that their set of nodes is a union of blocks. This is characteristic of weak cones, as can be checked.

Definition 14. Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be an adjunction structure and $\mathfrak{x} \in \mathfrak{C}$. The **constituent of $\mathfrak{S}$ headed by $\mathfrak{x}$** is the adjunction structure

$$\gamma(\mathfrak{x}) := \langle \downarrow \mathfrak{x}^{\circ}, \mathfrak{x}, <, \downarrow \mathfrak{x}\rangle$$

where $<$ is the restriction to $\downarrow \mathfrak{x}^{\circ}$ of $<$.

Now consider the second case, *remnant topicalization.* There is a rather popular analysis of argument ordering in German that relies on the assumption that the arguments of the verb are in canonical order at D–structure, while at S–structure they can practically appear in any order. This reordering is referred to as *scrambling.* Although we do not endorse the view that this is a correct analysis of the facts, let us assume that it is. Let us now look at the construction where the verb phrase is topicalized. This topicalization may happen after scrambling. Hence, in the extreme case it is the bare verb that moves, giving the the impression that we have an instance of head movement. (But since this is movement to specifier of comp, it is phrasal movement, and so it may not be analysed as head movement.)

(2.) *[Gelesen]$_1$ hat Alfred das Buch nicht* t_1.
Read has Alfred the book not.
'Alfred read this book not.'

(3.) *[Gelesen]$_1$ hat das Buch Alfred nicht* t_1.
Read has the book Alfred not.

(4.) *[Das Buch gelesen]$_1$ hat Alfred nicht* t_1.
The book read has Alfred not.

Suppose that only weak cones may move. Then scrambling cannot be adjunction to *VP*, as is often assumed. For then if the *VP* is moved, its scrambled arguments must move with it. So, after topicalization the verb appears at the beginning of the phrase with all its arguments — though they may occur in any order. Hence, we conclude that scrambling is either adjunction to a block containing the *VP* or else it is substitution. What it is depends on whether the chains formed by adjunction are A–chains or A'–chains. The literature varies considerably on this point. Some take scrambling to be substitution, some take it to be adjunction. If it is adjunction to *VP*, however, then it cannot be weak cones that move. Then we must conclude that there are two types of constituents, weak ones — if they are heads — and strong ones — if they are phrases. This is a rather unsatisfactory solution. Let us therefore adopt uniformly that constituents are weak. Scrambling is therefore substitution into a higher specifier position. Below we illustrate the relevant structures.

(5.) *Alfred$_1$ hat [[das Buch]$_2$ nicht [t$_1$[t$_2$gelesen]]]].*

(6.) *[t$_3$[t$_2$Gelesen]]$_1$ hat [[das Buch]$_2$ Alfred$_3$ nicht t$_1$].*

(7.) *[t$_3$[t$_2$Gelesen]]$_1$ hat [Alfred$_3$ [[das Buch]$_2$ nicht t$_1$].*

3.4. **Are Blocks Independent?** The next problem to be considered is whether blocks can be reconstructed given only the labelled tree. Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}, \ell\rangle$ be an adjunction structure. Put $\mathsf{Ntc}(\mathfrak{S}) := \langle S, r, <, \zeta\rangle$, where $\zeta(x) := \ell(b(x))$. We call $\mathsf{Ntc}(\mathfrak{S})$ the **node trace of** $\mathfrak{S}$. The question assumes the following form: do there exist nonisomorphic adjunction structures with identical node traces? If so, the node trace alone does not give enough information to let us recover the adjunction structure. The answer is positive, and counterexamples are easy to find (see Figs. 4 and 5 below). As we will see in this section the distinction between a base generated adjunct and a movement generated adjunct is lost by passing to the node trace. Therefore, the decomposition into blocks adds linguistically relevant distinctions. However, in order to establish the partition $\mathfrak{C}$ it is enough if we know only the upper segments of a category.

Proposition 15. *Let $\mathfrak{A} := \langle S, r, <, \mathfrak{C}\rangle$ be an adjunction structure. Put*

$$U(\mathfrak{C}) := \{\mathfrak{c}^\circ : \mathfrak{c} \in \mathfrak{C}\}$$

Then $\mathfrak{A}$ is uniquely identified by $\langle S, r, <\rangle$ and $U(\mathfrak{C})$.

Proof. Let $U := U(\mathfrak{C})$ be given. Define

$$x \sim_U y \quad \Leftrightarrow \quad (\forall u \in U)(x \leq u \leftrightarrow y \leq u)$$

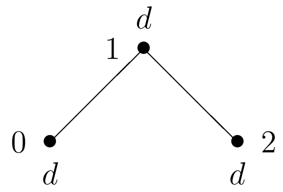
Put $\Pi := \{\{y : y \sim_U x\} : x \in S\}$. We show that $\Pi = \mathfrak{C}$. To that end, assume that for some $\mathfrak{x}$, $x, y \in \mathfrak{x}$. Then clearly $x \sim_U y$; for $x \leq \mathfrak{d}^\circ$ iff $\mathfrak{x}^\circ \leq \mathfrak{d}^\circ$ iff $y \leq \mathfrak{d}^\circ$. Conversely, assume that $x \sim_U y$. Put $\mathfrak{x} := b(x)$ and $\mathfrak{y} := b(y)$. Then $x \leq \mathfrak{x}^\circ$, and so $y \leq \mathfrak{x}^\circ$, since $x \sim_U y$. Thus, $\mathfrak{y}^\circ \leq \mathfrak{x}^\circ$. Hence $\mathfrak{x} \leq \mathfrak{y}$. Likewise $\mathfrak{x} \leq \mathfrak{y}$ is shown. Together this gives $\mathfrak{x} = \mathfrak{y}$, as desired. $\square$

Note by contrast that the lower segments are not sufficient for determining the blocks. Put

$$L(\mathfrak{C}) := \{\mathfrak{c}_\circ : \mathfrak{c} \in \mathfrak{C}\}$$

It turns out that in the picture below $L(\mathfrak{C}_2) = L(\mathfrak{C}_3)$. Again, notice that if 0 or 2 would be required to have different labels, there would be no question as how to reconstruct $\mathfrak{C}$. Moreover, consider the following restriction on blocks.

FIGURE 4. An Ambiguous Tree



Maximal Blocks. Blocks are maximal linear subsets of nodes with identical label.

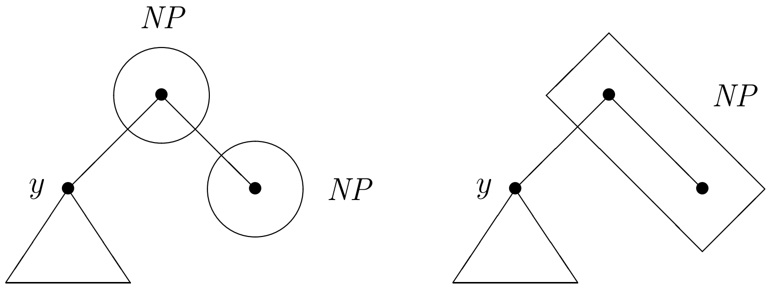
Often, structures are implicitly assumed to satisfy this property. Nevertheless, not only is there counterevidence, it is also not sufficient for the recovery of the blocks. For consider the following tree with $d \in D$ an arbitrary label. Without any restriction there exist three possible divisions into blocks (out of five possible partitions).

$$\begin{array}{rcl} \mathfrak{C}_1 & := & \{\{0\}, \{1\}, \{2\}\} \\ \mathfrak{C}_2 & := & \{\{0,1\}, \{2\}\} \\ \mathfrak{C}_3 & := & \{\{0\}, \{1,2\}\} \end{array}$$

Only $\mathfrak{C}_2$ and $\mathfrak{C}_3$ fulfill *Maximal Blocks*. However, we have no means to distinguish between them. Indeed, precisely this configuration poses problems in Chomsky's system of bare phrase structure, see Chomsky [5]. In general, this configuration also poses problems for the recovery of the head in the construction, since we assume that the head projects. (In the present context this means that it has the same label as the mother or is part of the same block as the mother.) So, the block with two segments corresponds to the head, to which the remaining block is adjoined. Fanselow [6] discussing such structures notes that they arise only through movement, so the head is nevertheless identifiable as the part that is not antecedent to a trace.

Interesting evidence against *Maximal Blocks* comes from the so–called *Split–DP* constructions. In the following sentences a part of an *DP* has been topicalized.

Figure 5. Base Generated versus Movement Generated Adjuncts



(8.) *[Teure klassische Bücher]$_1$ hat Alfred [viele t_1] gestohlen.*
Expensive classical books has Alfred many stolen.
'Alfred has stolen many expensive classical books.'

(9.) *[Klassische Bücher]$_1$ hat Alfed [viele teure t_1] gestohlen.*
Classical books has Alfred many expensive stolen.

(10.) *Bücher$_1$ hat Alfred [viele teure klassische t_1] gestohlen.*
Books has Alfred many expensive classical stolen.

In (8.) an entire *NP* is topicalized, in (9.) and (10.) however only parts of it. Since the movement is an instance of phrasal movement we must assume that in both cases an *NP* is moved. Hence, if this is an instance of movement at all — and there are reasons to believe that it is not — we must assume that the *NP* nodes are not part of the same block. In other words, we must assume that at D–structure each block is a singleton, and that proper blocks are created by adjunction movement. Althoug this weakens the appeal of the notion of adjunction put forward in [3], it has from time to time been noted that one must distinguish base generated adjuncts from movement generated adjuncts. The advantage of the adjunction structures is that this distinction can be made. Namely, a constituent headed by y is a *base generated adjunct* of x if y is a sister of x, and x and its mother z have identical label, but are in distinct blocks. The constituent headed by y is a *movement generated adjunct* of x if y is a sister of x and x is not an upper segment in its block. The picture below illustrates this distinction. To the left the constituent headed by y is a base generated adjunct, and to the right it is a movement generated adjunct.

4. Compression of Adjunction Structures

4.1. **Adjunction Trees.** An immediate question arises as to whether the two relations of inclusion and containment characterize an adjunction structure. Moreover, a related question is whether it is necessary that blocks are sets of nodes rather than a primitive entities. This makes sense especially since it is the basic idea of Chomsky in [3] that blocks should be the primitive objects, not the nodes contained in them.

Definition 16. An **adjunction tree** is a quadruple $\mathfrak{A} = \langle A, \mathfrak{a}, <, \ll \rangle$ such that

(a1) $\langle A, \mathfrak{a}, < \rangle$ is a tree,
(a2) $\langle A, \ll \rangle$ is a forest,
(a3) $< \subseteq \ll$,
(a4) if $x < y \ll z$ then $x \ll z$ as well, and
(a5) if $x < y < z$ and $x \ll z$ then also $y \ll z$.

Members of A are called **blocks**, $\mathfrak{a}$ is called the **root**, $<$ **containment relation** and $\ll$ the **inclusion relation**.

Definition 17. An **ordered** adjunction tree is a quintuple

$$\mathfrak{A} = \langle A, \mathfrak{a}, <, \ll, L \rangle$$

where $\langle A, \mathfrak{a}, <, \ll \rangle$ is an adjunction tree and L an ordering compatible with $\ll$.

The reader may first of all note that if $<$ is a tree ordering and $\ll \subseteq <$ then $\langle A, \ll \rangle$ is automatically a forest. So the second condition does not need to be checked if the others are satisfied.

Proposition 18. *Let* $\mathfrak{T} = \langle T, r, <, \mathfrak{C} \rangle$ *be an adjunction structure. Put* $\mathsf{T}(\mathfrak{T}) := \langle \mathfrak{C}, \mathfrak{r}, <, \ll \rangle$, *where* $r \in \mathfrak{r} \in \mathfrak{C}$. *Then* $\mathsf{T}(\mathfrak{T})$ *is an adjunction tree.*

Proof. We clearly have (a1), (a2) and (a3). For (a4), assume that $\mathfrak{a} < \mathfrak{b} \ll \mathfrak{c}$. Then $\mathfrak{b}^\circ < \mathfrak{c}_\circ$ and $\mathfrak{a}^\circ < \mathfrak{b}^\circ$. Then $\mathfrak{a}^\circ < \mathfrak{c}_\circ$, and so $\mathfrak{a} \ll \mathfrak{c}$. For (a5), assume that $\mathfrak{a} < \mathfrak{b} < \mathfrak{c}$ and $\mathfrak{a} \ll \mathfrak{c}$. Then $\mathfrak{a}^\circ < \mathfrak{b}^\circ < \mathfrak{c}^\circ$ and $\mathfrak{a}^\circ < \mathfrak{c}_\circ$. Since also $\mathfrak{b}^\circ > \mathfrak{a}^\circ$, $\mathfrak{b}^\circ$ is comparable with $\mathfrak{c}_\circ$. Clearly, since $\mathfrak{b}^\circ < \mathfrak{c}^\circ$, only $\mathfrak{b}^\circ < \mathfrak{c}_\circ$ can obtain. So, $\mathfrak{b} \ll \mathfrak{c}$. $\square$

Now let us look into the question of recovering the adjunction structure from a given adjunction tree. Let $\mathfrak{A} = \langle A, \mathfrak{a}, <, \ll \rangle$ be an adjunction tree. Say that $\mathfrak{b}$ is **adjoined to** $\mathfrak{c}$ if (1) $\mathfrak{b} < \mathfrak{c}$, (2) not $\mathfrak{b} \ll \mathfrak{c}$, (3) for no $\mathfrak{d}$, $\mathfrak{b} < \mathfrak{d} < \mathfrak{c}$. Let $Adj(\mathfrak{c})$ be the set of nodes adjoined to $\mathfrak{c}$. Another way of defining this set is by

$$Adj(\mathfrak{c}) = max_<(\downarrow \mathfrak{c} - \twoheaddownarrow \mathfrak{c})$$

Now form $A(\mathfrak{c}) := Adj(\mathfrak{c}) \times \{\mathfrak{c}\} = \{\langle \mathfrak{b}, \mathfrak{c}\rangle : \mathfrak{b} \in Adj(\mathfrak{C})\}$, and $N(\mathfrak{c}) := \{\mathfrak{c}_*\} \cup A(\mathfrak{c})$, where $\mathfrak{c}_* := \langle \mathfrak{c}, *\rangle$ for some suitable $*$. Finally,

$$N(\mathfrak{A}) := \bigcup_{\mathfrak{c} \in A} N(\mathfrak{c})$$

Moreover, $A_* := \{\mathfrak{c}_* : \mathfrak{c} \in A\}$. This will be the set of nodes of the adjunction structure. To define a tree ordering, choose for each $\mathfrak{c} \in A$ a linear ordering on $A(\mathfrak{c})$. We denote this ordering by $\lhd$. Next define Δ_i, $i < 6$, as follows.

$$\begin{array}{lll} x \Delta_0 y & \text{iff} & x = \mathfrak{c}_*, y = \langle \mathfrak{d}, x\rangle \\ x \Delta_1 y & \text{iff} & x = \mathfrak{c}_*, y \in \mathfrak{d}_*, \mathfrak{c} \ll \mathfrak{d} \\ x \Delta_2 y & \text{iff} & x = \mathfrak{b}_*, y = \langle \mathfrak{c}, \mathfrak{d}\rangle, \mathfrak{b} \ll \mathfrak{c} \text{ or } \mathfrak{b} = \mathfrak{c} \\ x \Delta_3 y & \text{iff} & x = \langle \mathfrak{c}, \mathfrak{d}\rangle, y = \mathfrak{e}, \mathfrak{d} \ll \mathfrak{e} \\ x \Delta_4 y & \text{iff} & x = \langle \mathfrak{c}, \mathfrak{d}\rangle, y = \langle \mathfrak{d}, \mathfrak{e}\rangle \\ x \Delta_5 y & \text{iff} & x = \langle \mathfrak{d}, \mathfrak{c}\rangle, y = \langle \mathfrak{e}, \mathfrak{c}\rangle, x \lhd y \end{array}$$

Now let

$$\begin{array}{lll} \nabla & := & \bigcup_{i<6} \Delta_i \\ < & := & \nabla^+ \end{array}$$

So, $<$ is the transitive closure of the union of the Δ_i. $\mathfrak{a}$ is the root in $\mathfrak{A}$. If $Adj(\mathfrak{a}) = \varnothing$ put $r(\mathfrak{A}) := \mathfrak{a}_*$. Otherwise, let $\langle \mathfrak{c}, \mathfrak{a}\rangle$ be the $\lhd$–maximal element in $A(\mathfrak{a})$ and put $r(\mathfrak{A}) := \langle \mathfrak{c}, \mathfrak{a}\rangle$. Finally,

$$\mathsf{A}(\mathfrak{A}, \lhd) := \langle N(\mathfrak{A}), r(\mathfrak{A}), <, \{N(\mathfrak{c}) : \mathfrak{c} \in A\}\rangle$$

Theorem 19. *Let $\mathfrak{A}$ be an adjunction tree. Then $\mathsf{A}(\mathfrak{A}, \lhd)$ is an adjunction structure.*

Proof. The relation $<$ is transitive by definition. We will prove at the end that $<$ is a tree ordering. Then, as can be checked, there is no element y such that $r(\mathfrak{A}) \nabla y$. Hence, $r(\mathfrak{A})$ is maximal. Now let $x \neq r(\mathfrak{A})$ be any element that is maximal. Then if $x = \langle \mathfrak{c}, \mathfrak{d}\rangle$, $\langle \mathfrak{c}, \mathfrak{d}\rangle$ must be maximal with respect to $\lhd$ in $A(\mathfrak{d})$. Furthermore, $\mathfrak{d}$ may not be adjoined to any block $\mathfrak{e}$, otherwise $\langle \mathfrak{c}, \mathfrak{d}\rangle < \langle \mathfrak{d}, \mathfrak{e}\rangle$, nor may $\mathfrak{d}$ be included in a block $\mathfrak{e}$, otherwise $\langle \mathfrak{c}, \mathfrak{d}\rangle < \mathfrak{e}$. Hence $\mathfrak{d}$ is $<$–maximal, that is, $\mathfrak{d} = \mathfrak{a}$. It follows that $x = r(\mathfrak{A})$. Now suppose that $x = \mathfrak{c}_*$. Then $A(\mathfrak{c}) = \varnothing$, and so $x = r(\mathfrak{A})$.

Now for the fact that $<$ is a tree ordering. We shall proceed as follows. First of all we notice that all Δ_i are irreflexive. Therefore ∇ is irreflexive. For every $x \neq r(\mathfrak{A})$ we show that there exists a y such that $x \nabla y$ and such that for all z with $x \nabla z$ we have $y < z$. We call this property of y with respect to x simply (‡). It follows first of all that $x \neq y$, and that that positions are linear. Now let $x \neq r(\mathfrak{A})$. Case 1. $x = \mathfrak{c}_*$. (Case 1A) Suppose that $Adj(\mathfrak{c}) \neq \varnothing$. Then $A(\mathfrak{c})$ is not empty. Let y be $\lhd$–minimal in it. Then y satisfies (‡). Proof. Let $x\Delta_i z$ for some $i < 6$. Then $i < 4$. Suppose that $i = 0$. Then $z = \langle \mathfrak{d}, \mathfrak{c}\rangle$ for some $\mathfrak{d}$. Hence $y = z$ or $y \lhd z$, and so $y \leq z$. Let $i = 1$. Then $z = \mathfrak{g}_*$ and $y\Delta_3 z$, hence $y < z$. Let $i = 2$. Then $z = \langle \mathfrak{d}, \mathfrak{e}\rangle$ such that $\mathfrak{c} \ll \mathfrak{d}$ and $y = \langle \mathfrak{f}, \mathfrak{c}\rangle$ for some $\mathfrak{f}$. Then $y\Delta_3\mathfrak{d}$ and $\mathfrak{d}\Delta_2 z$. So, $y < z$. (Case 1B) Suppose that $Adj(\mathfrak{c}) = \varnothing$. Then let $\mathfrak{d}$ be the least node with respect to $<$ such that $\mathfrak{c} < \mathfrak{d}$. It exists and is unique. If $\mathfrak{c} \ll \mathfrak{d}$, put $y := \mathfrak{d}_*$. Otherwise put $y := \langle \mathfrak{c}, \mathfrak{d}\rangle$. Then y satisfies (‡). Proof. Let $x\Delta_i z$. Then $i \in \{1, 2\}$. Suppose $i = 1$. Then, as $x < z$, we have $y \leq z$. Suppose $i = 2$. Then, $z = \langle \mathfrak{f}, \mathfrak{g}\rangle$ and $\mathfrak{c} \ll \mathfrak{f}$ or $x = \mathfrak{f}$. If $\mathfrak{c} \ll \mathfrak{f}$ then $\mathfrak{c} < \mathfrak{f}$ and so $\mathfrak{d} \leq \mathfrak{f}$. Therefore either $\mathfrak{d} = \mathfrak{f}$ or $\mathfrak{d} \ll \mathfrak{f}$, by (a5). From this $y \leq z$ follows. If $x = \mathfrak{f}$, $\mathfrak{g}$ is unique, and $z = y$. Case 2. $x = \langle \mathfrak{c}, \mathfrak{d}\rangle$ for some $\mathfrak{c}$ and $\mathfrak{d}$. (Case 2A) x is not $\lhd$–maximal in $A(\mathfrak{d})$. Then there exists by linearity of $\lhd$ a least element $\langle \mathfrak{b}, \mathfrak{d}\rangle$ in $A(\mathfrak{d})$ greater than x. Put $y := \langle \mathfrak{b}, \mathfrak{d}\rangle$. We show (‡) for y. Proof. Assume that $x\Delta_i z$. Then $i \in \{3, 4, 5\}$. Let $x\Delta_3 z$. Then $y\Delta_3 z$ as well. Let $x\Delta_4 z$. Then $z = \langle \mathfrak{d}, \mathfrak{e}\rangle$ for some $\mathfrak{e}$ and so $y\Delta_4 z$ as well. Finally, let $x\Delta_5 z$. Then $z = \langle \mathfrak{a}, \mathfrak{c}\rangle$ for some $\mathfrak{a}$, and by choice of y, either $\mathfrak{a} = \mathfrak{b}$ and so $y = z$ or $y \lhd z$, which gives $y\Delta_5 z$. (Case 2B) x is maximal in $A(\mathfrak{d})$. Let $\mathfrak{e}$ be the least $\mathfrak{f}$ such that $\mathfrak{f} > \mathfrak{d}$. $\mathfrak{e}$ is uniquely determined, since $<$ is a tree ordering on $\mathfrak{A}$. (Case 2Bα) $\mathfrak{d}$ is adjoined to $\mathfrak{e}$. Then $\langle \mathfrak{d}, \mathfrak{e}\rangle \in N(\mathfrak{A})$. In this case $y := \langle \mathfrak{d}, \mathfrak{e}\rangle$. We show that y satisfies (‡). Proof. First of all, $x\Delta_4 y$. Now let $x\Delta_i z$. Then $i \in \{3, 4\}$, the case $i = 5$ cannot arise. Assume $i = 3$. Then $z = \mathfrak{g}_*$ and $\mathfrak{d} \ll \mathfrak{g}$. Now $\mathfrak{c} < \mathfrak{d} < \mathfrak{e}$. Since $\mathfrak{e} \neq \mathfrak{g}$, we have $\mathfrak{e} < \mathfrak{g}$. Moreover, $\mathfrak{d} \ll \mathfrak{g}$ and so since $\mathfrak{e} \ll \mathfrak{g}$, by $(a5)$. Hence $y\Delta_3 z$, therefore $y < z$. Assume $i = 4$. Then $z = \langle \mathfrak{d}, \mathfrak{g}\rangle$ for some $\mathfrak{g}$. By choice of $\mathfrak{d}$, $\mathfrak{g}$ and $\mathfrak{e}$, $\mathfrak{g} = \mathfrak{e}$ and so $z = y$. (Case 2Bβ) $\mathfrak{d}$ is included in $\mathfrak{e}$. Then put $y := \mathfrak{e}_*$. We show (‡) for y. Proof. Let $x\Delta_i z$. Then only $i = 4$ can arise. So, $z = \mathfrak{f}_*$ and $\mathfrak{d} \ll \mathfrak{f}$. By definition, $\mathfrak{e} \leq \mathfrak{f}$. If $\mathfrak{e} = \mathfrak{f}$ we are done. Assume therefore $\mathfrak{e} < \mathfrak{f}$. Then, as $\mathfrak{d} < \mathfrak{e} < \mathfrak{f}$ and $\mathfrak{d} \ll \mathfrak{f}$ we also have $\mathfrak{e} \ll \mathfrak{f}$, by $(a4)$. So, $y\Delta_4 z$, concluding the proof. □

Not all adjunction structures are obtained in this way. What we get are structures in which nonminimal segments are binary branching.

Definition 20. Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be an adjunction structure. $\mathfrak{S}$ is called **natural** if for every block $\mathfrak{c}$ each $x \in \mathfrak{c} - \{\mathfrak{c}_\circ\}$ is exactly binary branching.

Theorem 21. *(1.) Let $\mathfrak{A} = \langle A, \mathfrak{a}, <, \ll\rangle$ be an adjunction tree. Let $\lhd$ be a family of linear orderings on the sets $A(\mathfrak{c})$, $\mathfrak{c} \in A$. Then $\mathfrak{A}$ is isomorphic to $\mathsf{T}(\mathsf{A}(\mathfrak{A}, \lhd))$. (2.) Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be a natural adjunction structure and $\lhd$ a family of linear orderings on $A(\mathfrak{c})$, $\mathfrak{c} \in \mathfrak{C}$,* *defined in the following way.* $\langle \mathfrak{a}, \mathfrak{c}\rangle \lhd \langle \mathfrak{b}, \mathfrak{c}\rangle$ *iff there exists a* $y \in \mathfrak{c}$ *such that* $\mathfrak{a}^\circ < y$ *but not* $\mathfrak{b}^\circ < y$*. Then* $\mathfrak{S}$ *is isomorphic to* $\mathsf{A}(\mathsf{T}(\mathfrak{S}), \lhd)$.

Proof. To show (1.), put $\phi : N(\mathfrak{c}) \mapsto \mathfrak{c}$. It turns out that $\mathfrak{c}_* < \mathfrak{d}_*$ is equivalent to $\mathfrak{c} \ll \mathfrak{d}$ (by definition of Δ_1 and the properties of adjunction trees). *ϕ is bijective.* For if $N(\mathfrak{c}) \neq N(\mathfrak{d})$ then also $\mathfrak{c} \neq \mathfrak{d}$. Moreover, $\mathfrak{c}$ is the image of $N(\mathfrak{c})$. *ϕ respects roots.* We have put $r(\mathfrak{A}) = \mathfrak{a}$ if $\mathfrak{a}$ is not adjoined to, and $r(\mathfrak{a}) = \langle \mathfrak{a}, \mathfrak{x}\rangle$ for the highest adjunct $\mathfrak{x}$. In both cases, $\phi(r(\mathfrak{A})) = \mathfrak{a}$. *$\phi$ respects $\ll$.* Notice first of all that $\mathfrak{c}$ is the smallest element in $N(\mathfrak{c})$ with respect to $<$. Hence $N(\mathfrak{c}) \ll N(\mathfrak{d})$ iff $\mathfrak{c} < \mathfrak{d}$, by Proposition 5, (3.). The latter is equivalent to $\mathfrak{c} \ll \mathfrak{d}$, as desired. *$\phi$ respects $<$.* For the relation $<$, several cases have to be distinguished. $N(\mathfrak{c}) < N(\mathfrak{d})$ iff there is a $x \in N(\mathfrak{c})$ and a $y \in N(\mathfrak{d})$ such that $x < y$. Indeed, we can actually assume that $x \prec y$ and therefore $x\Delta_i y$ for some i. $\underline{i = 0}$. Cannot arise, since $\mathfrak{c} \neq \mathfrak{d}$. $\underline{i = 1}$. Then $y = \mathfrak{d}_*$, $x = \mathfrak{c}_*$ and $\mathfrak{c} \ll \mathfrak{d}$ and so $\mathfrak{c} < \mathfrak{d}$. $\underline{i = 2}$. Then $y = \langle \mathfrak{b}, \mathfrak{d}\rangle$ for some $\mathfrak{b}$ and $x = \mathfrak{c}_*$, with $\mathfrak{c} = \mathfrak{b}$ or $\mathfrak{c} \ll \mathfrak{b}$. Since $\mathfrak{b} < \mathfrak{d}$ ($\mathfrak{b}$ is adjoined to $\mathfrak{d}$), we have $\mathfrak{c} < \mathfrak{d}$ in all cases. $\underline{i = 3}$. $y = \mathfrak{d}_*$ and $x = \langle \mathfrak{b}, \mathfrak{c}\rangle$ for some $\mathfrak{b}$ as well as $\mathfrak{c} \ll \mathfrak{d}$. Hence $\mathfrak{c} < \mathfrak{d}$. $\underline{i = 4}$. $y = \langle \mathfrak{c}, \mathfrak{d}\rangle$ and $x = \langle \mathfrak{a}, \mathfrak{c}\rangle$ for some $\mathfrak{a}$. Then $\mathfrak{c} < \mathfrak{d}$. $\underline{i = 5}$. Cannot arise, since $\mathfrak{c} \neq \mathfrak{d}$. The reasoning can be played in the converse direction. Then we assume that $\mathfrak{c} \prec \mathfrak{d}$ and show that $N(\mathfrak{c}) < N(\mathfrak{d})$. This is straightforwardly verified. This finishes the proof of the first claim.

Let us now show (2.). Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be given and a family of linear orders on $N(\mathfrak{c})$, $\mathfrak{c} \in \mathfrak{C}$, such that $\langle \mathfrak{a}, \mathfrak{c}\rangle \lhd \langle \mathfrak{b}, \mathfrak{c}\rangle$ if there is a $y \in \mathfrak{c}$ such that $\mathfrak{a}^\circ < y$ but not $\mathfrak{b}^\circ < y$. Assume also that $\mathfrak{S}$ is natural. We define $\psi : N(T(\mathfrak{S})) \to S$ as follows. Put $\psi : \mathfrak{c}_* \mapsto \mathfrak{c}_\circ$, and let $\langle \mathfrak{c}, \mathfrak{d}\rangle \in N(\mathfrak{d})$. Let z be the least node in S dominating both $\mathfrak{c}_\circ$ and $\mathfrak{d}_\circ$. Then put $\psi(\langle \mathfrak{c}, \mathfrak{d}\rangle) := z$. *$\psi$ is bijective.* For if $x \in S$, then let $x \in \mathfrak{c}$. If x is the lowest segment in $\mathfrak{c}$ then $x = \psi(\mathfrak{c})$. If not, there exists a $\mathfrak{d}$ such that $\mathfrak{d}^\circ$ is a daughter of x which is not in $\mathfrak{c}$. $\mathfrak{d}$ exists and is unique, since $\mathfrak{S}$ is natural. Now $\mathfrak{d}$ is adjoined to $\mathfrak{c}$ and so $\langle \mathfrak{d}, \mathfrak{c}\rangle \in N(\mathfrak{c})$. Furthermore, $\psi(\langle \mathfrak{d}, \mathfrak{c}\rangle) = x$. Let ρ be the inverse of ψ. We show that ρ is bijective. So let $x \neq y$. Various cases arise. If x and y belong to different blocks, then clearly $\rho(x) \neq \rho(y)$. So let them be in the same block. If one of them is minimal, the other is not, and then $\rho(x) \neq \rho(y)$. So, let them both be nonminimal. Then $x = \langle \mathfrak{a}, \mathfrak{c}\rangle$ and $y = \langle \mathfrak{b}, \mathfrak{c}\rangle$ for certain distinct $\mathfrak{a}$, $\mathfrak{b}$ and $\mathfrak{c}$. So, in all cases $\rho(x) \neq \rho(y)$. It is clear that each element of $N(T(\mathfrak{S}))$ is of the form $\rho(x)$ for some x. *ψ respects roots.* Follows from the fact that ψ respects $<$ and the fact that the root $\mathsf{A}(\mathfrak{S}, \lhd)$ is the maximal segment of the block containing r. *ψ respects $<$.* We show that (i) if $x \prec y$ then $\rho(x)\nabla\rho(y)$. Namely, let $x \prec y$. Let $x \in \mathfrak{c}$ and

$y \in \mathfrak{d}$. Suppose first that $\mathfrak{c} = \{x\}$. Then let $y = \mathfrak{d}_\circ$. So $\rho(x) = \mathfrak{c}_*$ and $\rho(y) = \mathfrak{d}_*$. Furthermore, $\mathfrak{c} \ll \mathfrak{d}$ from which $\rho(x)\Delta_1\rho(y)$. If y is not minimal in $\mathfrak{d}$ then we get $\rho(x)\Delta_2\rho(y)$. Now suppose that $\mathfrak{c} \neq \{x\}$. Then two cases arise. (Case 1) x is maximal in $\mathfrak{c}$, and (Case 2) x is not maximal in $\mathfrak{c}$. In Case 2, $\mathfrak{c} = \mathfrak{d}$, and $\rho(x)\Delta_0\rho(y)$ if x is minimal in $\mathfrak{c}$ and $\rho(x)\Delta_5\rho(y)$ if not. (For then, $\rho(x) = \langle\mathfrak{a}, \mathfrak{c}\rangle$ and $\rho(y) = \langle\mathfrak{b}, \mathfrak{c}\rangle$, where $\rho(x) \lhd \rho(y)$. This means $\mathfrak{a}_* < \langle\mathfrak{a}, \mathfrak{c}\rangle$, but not $\mathfrak{b}_* < \langle\mathfrak{b}, \mathfrak{c}\rangle$.) In Case 1, we get $\rho(x)\Delta_4\rho(y)$ if y is not minimal in $\mathfrak{d}$, and $\rho(x)\Delta_3\rho(y)$ if y is minimal in $\mathfrak{d}$. This completes all cases. (ii) If $\rho(x)\nabla\rho(y)$ then $x < y$. This however is established by checking all cases. Hence the claim (2.) proved. □

Corollary 22. *Let $\mathfrak{A}$ be an adjunction tree. Call $\mathfrak{A}$* **rigid** *if for any normal adjunction structures $\mathfrak{S}_1$ and $\mathfrak{S}_2$ $\mathsf{T}(\mathfrak{S}_1)$ is isomorphic to $\mathsf{T}(\mathfrak{S}_2)$ iff $\mathfrak{S}_1$ is isomorphic to $\mathfrak{S}_2$. An adjunction tree is rigid iff to each block at most one block is adjoined.*

4.2. **The Ordered Case.** One expects that ordering will help in making the reconstruction of the adjunction structure unique. We will see that this holds only under certain conditions. First, let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be given. For $\mathfrak{a}$ and $\mathfrak{b}$ adjoined to $\mathfrak{c}$ we say that $\mathfrak{a}$ is **adjoined higher than** $\mathfrak{b}$ if there exists a $y \in \mathfrak{c}$ such that $\mathfrak{b}^\circ < y$ but it does not hold that $\mathfrak{a}^\circ \leq y$. We have seen that in compressing the adjunction structure we must put $\langle\mathfrak{b}, \mathfrak{c}\rangle \lhd \langle\mathfrak{a}, \mathfrak{c}\rangle$ iff $\mathfrak{a}$ is adjoined higher than $\mathfrak{b}$. Now let us introduce an ordering on $\mathfrak{S}$. We shall see that different orderings give rise to identical ordered adjunction trees. Notice first that given an adjunction tree, all that needs to be established is how a node of the tree is spelled out as a block with segments, and how the segments are ordered with respect to $\lhd$. What determines the shape of the block $N(\mathfrak{c})$ is the set of immediate daughters, $D(\mathfrak{c}) := max_<\{\mathfrak{d} : \mathfrak{d} < \mathfrak{c}\}$. $\mathfrak{a}$ is **adjoined to the left** (**right**) of $\mathfrak{c}$ if $\mathfrak{a}$ is adjoined to $\mathfrak{c}$ and $\mathfrak{a}\,L\,\mathfrak{c}$ ($\mathfrak{c}\,L\,\mathfrak{a}$). Let $LA(\mathfrak{c})$ ($RA(\mathfrak{c})$) be the set of blocks adjoined to the left (right) of $\mathfrak{c}$. Furthermore, let $IN(\mathfrak{c})$ be the immediately included blocks. It turns out that both sets are linearly ordered by $\lhd$. For let $\mathfrak{a}$ and $\mathfrak{b}$ be adjoined to the left to $\mathfrak{c}$ and $\mathfrak{a}$ higher than $\mathfrak{b}$. Let x, y and z be the following segments. Then $\mathfrak{a}^\circ\,L\,y$, $\mathfrak{b}^\circ\,L\,z$. Then $\mathfrak{a}^\circ\,L\,\mathfrak{b}^\circ$, by the fact that the order is compatible. So, if $\mathfrak{a}$ is higher and both are left adjoined, then $\mathfrak{a}$ is left of $\mathfrak{b}$. Dually, if both are right adjoined and $\mathfrak{a}$ is higher than $\mathfrak{b}$ then $\mathfrak{b}\,L\,\mathfrak{a}$. Now, the set $D(\mathfrak{c})$ of $<$–daughters of $\mathfrak{c}$ is partitioned into three sets, $LA(\mathfrak{c})$, $IN(\mathfrak{c})$ and $RA(\mathfrak{c})$. Each element of the first is left of every element of the second, and each element of the second is left of every element of the third. Now each element of $N(\mathfrak{c})$ is of the form $\langle\mathfrak{x}, \mathfrak{c}\rangle$ where $\mathfrak{x} \in LA(\mathfrak{c})$ or $\mathfrak{x} \in RA(\mathfrak{c})$. Choose two orderings $\lhd_1$ and $\lhd_2$ such

FIGURE 6. Two Left Adjuncts

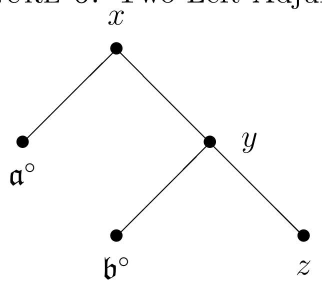
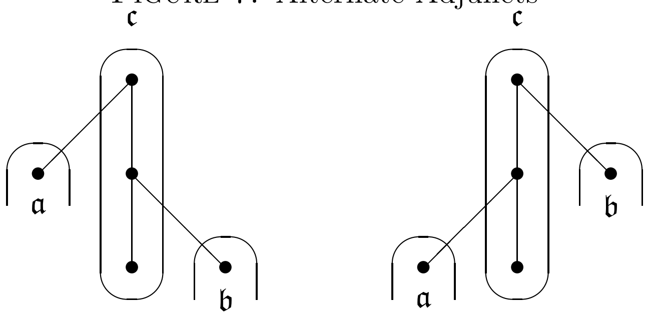


FIGURE 7. Alternate Adjuncts



that $LA(\mathfrak{c}) \times \{\mathfrak{c}\}$ as well as $RA(\mathfrak{c}) \times \{\mathfrak{c}\}$ are linearly ordered by $\lhd_1$ and $\lhd_2$. Then $\mathsf{T}(\mathsf{A}(\mathfrak{A}, \lhd_1)) \cong \mathsf{T}(\mathsf{A}(\mathfrak{A}, \lhd_2))$.

Definition 23. An ordered adjunction structure is **homogeneous** if for every node all nodes are adjoined to the left or all to the right.

Theorem 24. *An ordered adjunction tree is the compression of exactly one ordered adjunction structure iff it is homogeneous.*

We have already seen that homogeneous adjunction trees have the desired property. Now let $\mathfrak{A}$ be not homogeneous. Then there is a $\mathfrak{c}$, $\mathfrak{a}$ and $\mathfrak{b}$ such that $\mathfrak{a}$ is adjoined to the left of $\mathfrak{c}$ and $\mathfrak{b}$ to the right of $\mathfrak{c}$. Then we may define $\lhd_1$ such that $\mathfrak{a}$ is higher than $\mathfrak{b}$ and $\lhd_2$ such that $\mathfrak{b}$ is higher than $\mathfrak{a}$. Homogeneity is a consequence of a stronger requirement, namely *right* or *left uniformity.*

Definition 25. An ordered adjunction structure (adjunction tree) is **left uniform** (**right uniform**) if adjuncts are always left (right) of the blocks to which they are adjoined.

Haider [7] assumes that the basic projection line goes down if one moves left–to–right. The LCA of Kayne [8] (see below) has as a consequence that structures are left uniform. So, both Haider and Kayne assume that structures are left uniform. This has the advantage that from the adjunction tree the adjunction structure can be recovered. That both should converge on this is an accident. Both base their restriction on c–command as a diagnostic for constituent structure. C–command, however, used by Kayne is defined blockwise and does not allow to distinguish different adjuncts which are adjoined at the same side. Haider's notion is node based, and his arguments are derived from empirical observations concerning the sequence of arguments of a verb.

5. Domains

5.1. **Block Based Command Relations.** The change from trees to adjunction structures necessitated a revision of the notion of command relations. Command relations shall now be defined not nodewise but blockwise (so in fact we are defining them over adjunction trees).

Definition 26. Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C}\rangle$ be an adjunction structure and $r \in \mathfrak{r} \in \mathfrak{C}$. A **command relation** over $\mathfrak{S}$ is a binary relation R on $\mathfrak{C}$ such that (1.) $\mathfrak{r}R = \mathfrak{C}$, (2.) for $\mathfrak{y} \neq \mathfrak{r}$ the domain $\mathfrak{y}R$ is a constituent properly containing the constituent headed by $\mathfrak{y}$, and (3.) if $\mathfrak{x} \leq \mathfrak{y}$, then $\mathfrak{x}R \subseteq \mathfrak{y}R$. The **node–trace** of R is the set $\{\langle x, y\rangle : (\exists \mathfrak{c}, \mathfrak{d} \in \mathfrak{C})(x \in \mathfrak{c}, y \in \mathfrak{d} \textit{ and } \mathfrak{c}R\mathfrak{d})\}$.

There are four choices, depending on whether positions are weak or strong upper cones, and whether constituents are weak or strong lower cones. Namely, consider the following definition of O–command. x O–commands y iff for all z in the position of x with label O, y is in the constituent headed by z. Depending on the choice of weak and strong for positions and constituents we end up with different notions of a command relation. However, as we have noted earlier, constituents are weak cones. So, two of the four choices are excluded. We are thus left with two notions.

Let us call a command relation R **weak** if for the generating function f_R we have $f_R(\mathfrak{x}) \geq \mathfrak{x}$ for all $\mathfrak{x}$ and $f_R(\mathfrak{x}) > \mathfrak{x}$ for all $\mathfrak{x} < \mathfrak{r}$. R is **strong** if $f_R(\mathfrak{x}) \gg \mathfrak{x}$ for all $\mathfrak{x}$ such that $\mathfrak{x} \ll \mathfrak{r}$, and $f_R(\mathfrak{x}) = \mathfrak{r}$ otherwise. Then it is clear what we understand by **strong** O–command and **weak** O–command. Is there a way to choose between strong and weak relations? The answer is indirect. We will show that if we opt for weak relations, they turn out to be node based, so the entire reasons of introducing adjunction structures in [3] disappear. If we opt for the strong relations, however, we get the intended effects of the barriers system. Let us say a command relation on an adjunction structure is **node based** if the node–trace is a command relation. We want to require the following property.

Node Compatibility. A block based relation is a command relation only if its node trace is a command relation.

This excludes the relations where constituents are strong constituents. For the node trace of a strong constituent is not necessarily a constituent. For let $\mathfrak{c}$ be a two segment block. Assume that $\mathfrak{d}$ is adjoined to $\mathfrak{c}$. Then the strong cone of $\mathfrak{c}$ does not contain $\mathfrak{d}$. However, $\mathfrak{c}^\circ$ is a member of the node trace of $\downarrow \mathfrak{c}$. So, the node trace of $\downarrow \mathfrak{c}$ is not a constituent.

We now want to compute the node trace of the command relations of weak and strong O–command. Let $O \subseteq D$ be a set of labels. Then let $\mathsf{O}_\bullet$ be the (node based) command relation based on the set of nodes which have label from O and are minimal segments in their block. Let $\mathsf{O}^\bullet$ be the (node based) command relation based on the set of nodes that have label O and are maximal segments. Finally, let O_μ be the relation based on the nodes with label O which are both maximal and minimal. Denote by $W(O)$ the node trace of weak O–command, and by $S(O)$ the node trace of strong O–command.

Theorem 27.
$$\begin{array}{rcl} W(O) & = & \mathsf{O}^\bullet \\ S(O) & = & (\mathsf{O}_\bullet \circ \mathsf{O}^\bullet) \cap \mathsf{O}_\mu \end{array}$$
In particular, $W(O)$ is tight.

Proof. Let $\langle x, y\rangle \in W(O)$ and let $x \in \mathfrak{x}$ and $y \in \mathfrak{y}$. Then for the least $\mathfrak{z}$ which is $> \mathfrak{x}$ and has label in O, $\mathfrak{z} \geq \mathfrak{y}$ — if it exists. Assume $\mathfrak{z}$ exists. Then $\mathfrak{z}^\circ$ is also the least node with label in O which is a maximal segment and $> \mathfrak{x}^\circ$. It follows that $\mathfrak{z}^\circ > x$ and also that $\mathfrak{z}^\circ \geq y$. Hence, $\langle x, y\rangle \in \mathsf{O}^\bullet$. Assume that $\mathfrak{z}$ does not exist. Then no block above $\mathfrak{x}$ has a label in O. Then no node above x has a label from O, and so $\langle x, y\rangle \in W(O)$. Conversely, let $\langle x, y\rangle \in \mathsf{O}^\bullet$. Let z be the least node with label from O which is a maximal segment. (If it does not exist, we are done, as can be seen easily.) Let $\mathfrak{z}$ be its block. Then $\mathfrak{z}$ is the least block $> \mathfrak{x}$ with label from O. So, $\mathfrak{x}$ weakly O–commands $\mathfrak{y}$. Hence $\langle x, y\rangle \in W(O)$. Now for $S(O)$. Let $\langle x, y\rangle \in S(O)$. Let $x \in \mathfrak{x}$ and $y \in \mathfrak{y}$. Assume there exists a block $\gg \mathfrak{x}$ with a label from O. Let $\mathfrak{z}$ be the least such block. Two cases arise. (Case 1.) $\mathfrak{z}^\circ = \mathfrak{z}_\circ = z$. Then z is minimal and maximal at the same time. Then $x < z$ and $y \leq z$. It is then clear that $\langle x, y\rangle \in \mathsf{O}_\mu$. Moreover, $\langle x, y\rangle \in \mathsf{O}_\bullet$ and so $\langle x, y\rangle \in \mathsf{O}_\bullet \circ \mathsf{O}^\bullet$. (Case 2.) $\mathfrak{z}_\circ < \mathfrak{z}^\circ$. Then $\langle x, \mathfrak{z}^\circ\rangle \in \mathsf{O}_\bullet \circ \mathsf{O}^\bullet$ but also $\langle x, \mathfrak{z}^\circ\rangle \in \mathsf{O}_\mu$. Thus, $\langle x, y\rangle \in (\mathsf{O}_\bullet \circ \mathsf{O}^\bullet) \cap \mathsf{O}_\mu$. Now assume conversely that $\langle x, y\rangle \in (\mathsf{O}_\bullet \circ \mathsf{O}^\bullet) \cap \mathsf{O}_\mu$. Let $x \in \mathfrak{x}$, $y \in \mathfrak{y}$. Let $\mathfrak{z}$ be the least block of label from O such that $\mathfrak{z} \gg \mathfrak{x}$. Then $\mathfrak{z} \geq \mathfrak{y}$. This shows the theorem. $\Box$

As before, we can use a generating function for command relations. A block based command relation R is tight if it satisfies the postulates for tight relations.

Tightness. If $f_R(\mathfrak{x})$ is in the position of $\mathfrak{y}$ then $f_R(\mathfrak{y}) = f_R(\mathfrak{x})$ or $f_R(\mathfrak{x})$ is also in the position of $f_R(\mathfrak{y})$.

Clearly, among the tight relations we are interested in the analogues of $\kappa(O, \mathfrak{S})$ for $O \subseteq D$. We put $\langle \mathfrak{x}, \mathfrak{y} \rangle \in \lambda(O, \mathfrak{S})$ iff for all $\mathfrak{z} \gg \mathfrak{x}$ such that $\ell(\mathfrak{z}) \in O$ we have $\mathfrak{y} \leq \mathfrak{z}$.

Definition 28. Let $\mathfrak{S}$ be a labelled (ordered) adjunction structure with labels over D. A **definable** command relation over $\mathfrak{S}$ is a command relation generated from relations of the form $\lambda(O, \mathfrak{S})$, $O \subseteq D$, using intersection and relation composition.

5.2. **K(ayne)–Structures.** In [8], Kayne proposes a constraint on adjunction structures which he calls the *Linear Correspondency Axiom* (LCA). This axiom connects precedence with antisymmetric c–command. Here, x c–commands y antisymmetrically if x c–commands y but y does not c–command x. Kayne's theory is illustrative for the potential of adjunction structures, yet also for its dangers. The attractiveness of the proposal itself — namely, to link precedence with hierarchy — disappears as soon as one starts to look at the details. For the canonical definition of c–command does not yield the intended result. It is too restrictive. Hence, to make the theory work, a new definition has to be put in place of it, that takes constituents and positions to be strong. Although it too is restrictive (so that in the book Kayne has to go through many arguments to show that it is the right theory) it resorts to a definition of c–command that we have actually discarded on theory internal reasons, since it takes the wrong notion of a constituent.

Definition 29. x **ac–commands** y if x and y do not overlap and x c–commands y but y does not c–command x.

A nice characterization of c–command and ac–command can be given in the following way. Let $\mu(x)$ be the mother of x. This is undefined if x is the root.

Lemma 30. *(1) x c–commands y iff x is the root or $\mu(x) \geq y$. (2) x ac–commands y iff (a) x and y do not overlap and (b) $\mu(x) > \mu(y)$.*

Proof. (1) Suppose that x c–commands y. If x is not the root, $\mu(x)$ is defined and $\mu(x) \geq y$, by definition of c–command. The converse is also straightforward. (2) Suppose that x ac–commands y. Then neither x nor y can be the root. Then $\mu(x) \geq y$. Now, $\mu(x) = y$ or $\mu(x) > y$. $\mu(x) = y$ cannot hold, since then y overlaps with x. So $\mu(x) > y$ and hence $\mu(x) \geq \mu(y)$. However, $\mu(x) = \mu(y)$ implies that y c–commands x, which is excluded. So, $\mu(x) > \mu(y)$. Conversely, assume that x and y do not overlap and that $\mu(y) < \mu(x)$. Then neither is the root and $\mu(x) > y$, since $y \leq \mu(y)$. So, x c–commands y. If y c–commands x then $\mu(y) \geq x$, which in combination with $\mu(y) < \mu(x)$ gives $\mu(y) = x$. This is excluded. Hence y does not c–command x. □

Proposition 31. *Antisymmetric c–command is irreflexive and transitive.*

Proof. Irreflexivity follows immediately from the definition. Suppose that x ac–commands y and that y ac–commands z. Clearly, none of the three is the root of the tree. Then $\mu(x) > \mu(y) > \mu(z)$, from which $\mu(x) > \mu(z)$. Now suppose that x overlaps with z. $x \leq z$ cannot hold, for then $\mu(x) \leq \mu(z)$. So, $x > z$. Hence $\mu(y)$ must overlap with x, for also $\mu(y) > z$. Since $\mu(x) > \mu(y)$, we have $x \geq \mu(y)$ and so $x > y$. This is a contradiction, for x does not overlap with y. □

We will suspend the full definition of ac–command for adjunction structures and state first the LCA. After having worked out the consequences of the LCA for trees we will return to the definition of ac–command.

Definition 32. Let $\mathfrak{S} = \langle S, r, <, \mathfrak{C} \rangle$ be an adjunction structure. Define a binary relation on $\mathfrak{C}$, $\rtimes$, by $\mathfrak{c} \rtimes \mathfrak{d}$ iff there exists $\mathfrak{x} \geq \mathfrak{c}$ and $\mathfrak{y} \geq \mathfrak{d}$ such that $\mathfrak{x}$ ac–commands $\mathfrak{y}$.

Thus the LCA can be phrased in the following form.

Linear Correspondence Axiom. $\rtimes$ is a linear order on the leaves.

We remark that $\rtimes$ depends on the particular choice of the notion of c–command. We will play with several competing definitions and see how LCA constrains the structure of adjunction structures depending on the particular definition of c–command. Let us give a special name for the structures satisfying LCA.

Definition 33. Let $\mathfrak{S}$ be an adjunction structure, and $X \subseteq S^2$ a binary relation over S. Put

$$\rtimes_X := \{\langle \mathfrak{x}, \mathfrak{y} \rangle : (\exists \mathfrak{v} \geq \mathfrak{x})(\exists \mathfrak{w} \geq \mathfrak{y})(\langle \mathfrak{v}, \mathfrak{w} \rangle \in X)\}$$

$\mathfrak{S}$ is called a **K(X)–structure** if $\rtimes_X$ is a linear order on the leaves.

Recall that a leaf is a block that contains a segment which is a leaf of the underlying tree. First we take X to be the notion of ac–command defined above. The result is quite dissimilar to those of Kayne [8]. The reason is that Kayne chooses a different notion, which we call sc–command. It is defined below.

To see how K(AC)–structures look like, let us start with a limiting case, namely trees. Notice that in general, if x ac–commands y and $y \geq z$ then x ac–commands z as well.

Definition 34. Let $\mathfrak{T} = \langle T, r, < \rangle$ be a tree. $\mathfrak{T}$ is called a **K(AC)–tree** if $x \rtimes y$ is linear on the leaves, where $x \rtimes y$ iff there exists $v \geq x$, $w \geq y$ such that v ac–commands w.

Theorem 35 (Kayne)**.** *A tree is a K(AC)–tree iff it is at most binary branching, and for every x, y_1, y_2, $y_1 \neq y_2$ such that $y_1 \prec x$ and $y_2 \prec x$, either y_1 is a leaf or y_2 is a leaf, but not both.*

Proof. Let $\mathfrak{T}$ be a tree. Let x and y be leaves. We claim that $x \rtimes y$ and *not* $y \rtimes x$ iff x ac–commands y. Assume that $x \rtimes y$ and not $y \rtimes x$. Then x and y do not overlap. (Otherwise, let $u \geq x$ and $v \geq y$. Then u does not ac–command v since u and v also overlap.) If x does not ac–command y then either (Case 1) x does not c–command y or (Case 2) y c–commands x. Case 2 is easily excluded. For then $x \rtimes y$ simply cannot hold since every node dominating y must c–command x. Suppose Case 1 obtains. By definition of $\rtimes$ and the remark preceding Definition 34 there is a u such that $x \leq u$ and u ac–commands y but $u \not\geq y$. Moreover, since y does not c–command x there is a $v > y$ such that v c–commands x but $v \not\geq x$. Hence, x does not c–command v, otherwise x c–commands y. Hence, v ac–commands x. We now have: u ac–commands y, whence $x \rtimes y$, and v ac–commands x, whence $y \rtimes x$. Contradiction. So, x ac–commands y. If x ac–commands y then by definition $x \rtimes y$. Moreover, if for some $v \geq y$ and some $u \geq x$ we have that v ac–commands u, then $v \not\geq x$ and $u \not\geq y$, from which follows that $u = x$, and u c–commands v, a contradiction. This proves our claim. Suppose now that $\mathfrak{T}$ is a K(AC)–tree. Let x be a node with three (pairwise distinct) daughters, y_1, y_2 and y_3. Let z_i ($i \in \{1, 2, 3\}$) be leaves such that $z_i \leq y_i$ for all i. By LCA, the z_i are linearly ordered by $\rtimes$. Without loss of generality we assume $z_1 \rtimes z_2 \rtimes z_3$. Then z_1 ac–commands z_2 and z_2 ac–commands z_3. Therefore, $z_1 = y_1$ and $z_2 = y_2$. But then z_2 ac–commands z_1, a contradiction. So, any node has at most two daughters. Likewise, if x has exactly two daughters then exactly one must be a leaf. Now assume that $\mathfrak{T}$ satisfies all these requirements. We will show that it is a K(AC)–tree. First, any node is either a leaf or a mother of a leaf. Mothers of leaves are linearly ordered by $>$. (For if not, take mothers u and v that do not overlap. Let w be their common ancestor. w has at least two daughters of which neither is a leaf.) Now let $x \rtimes y$. Let u be the least node dominating x and y. Then, u has two daughters, of which one is a leaf. This is easily seen to be x. Further, y is not a leaf. So, $y \rtimes x$ cannot obtain. Hence we have $x \rtimes y$ iff x ac–commands y. This is irreflexive and transitive. Now, finally, take a leaf x. It has a mother $\mu(x)$ (unless the tree is trivial). From what we have established, $x \rtimes y$ for a leaf y iff $\mu(x) > \mu(y)$. But the mothers of leaves are linearly ordered by $>$, as we have seen. $\Box$

A few comments are in order. If ac–command would not require x and y to be incomparable then x ac–commands y if $x > z > y$ for some z. Then trees satisfying LCA would have height 2 at most. If we would instead use the relation of ec–command, where x ec–commands y if x c–commands y but $x \not\geq y$ then x ac–commands y if $x \prec y$. So, trees satisfying LCA would again be rather flat. We would achieve the same result as above in Theorem 35, however, if the definition of c–command would be further strengthened as follows. x cc–commands y if x and y are incomparable and x c–commands y; x acc–commands y if x cc–commands y but y does not cc–command x. (Simply note that acc–command is the same relation as ac–command.)

Now let us go over to adjunction structures. The results we are going to provide shall in the limiting case of a tree return the characterizations above.

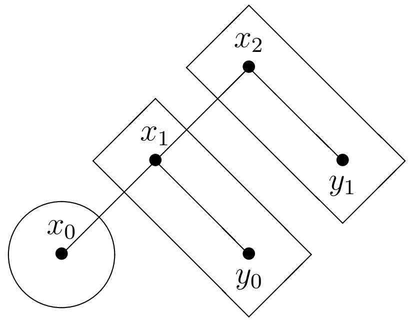
Definition 36. In an adjunction structure, $\mathfrak{x}$ **wc–commands** $\mathfrak{y}$ if $\mathfrak{x} \not\geq \mathfrak{y}$ and for every block $\mathfrak{u} \gg \mathfrak{x}$ we have $\mathfrak{u} \geq \mathfrak{y}$.

This is c–command as defined earlier for blocks, with the added condition that $\mathfrak{x}$ excludes $\mathfrak{y}$. In the tree case this is like c–command, but the clause $x \not\geq y$ is added.

Definition 37. In an adjunction structure, $\mathfrak{x}$ **awc–commands** $\mathfrak{y}$ if $\mathfrak{x}$ and $\mathfrak{y}$ are $\ll$–incomparable, $\mathfrak{x}$ wc–commands $\mathfrak{y}$ but $\mathfrak{y}$ does not wc–command $\mathfrak{x}$.

This is the same as ac–command in the tree case, hence the results carry over. First of all, we assume that adjunction structures do not contain segments which are nonbranching and nonminimal. Recall that a morphological head of an adjunction structure is a maximal constituent in which no block includes another block. With respect to heads, the LCA is less strict on adjunction structures. The reason is that the exlusion of overlap is replaced by the condition that no

FIGURE 8. A K(AWC)–Structure which is a Head



block includes the other. It turns out that in adjunction structures the following type of morphological heads are admitted.

Theorem 38. *Let $\mathfrak{A}$ be a morphological head and a K(AWC)–structure. Then $\mathfrak{A}$ is right branching. That is to say, $\mathfrak{A} = \langle T, <, r, L, \mathfrak{C}\rangle$ where*

$$\begin{array}{rcl} T & := & \{x_i : i < n+1\} \cup \{y_i : i < n\} \\ < & := & \{\langle x_i, x_j\rangle : i < j < n+1\} \cup \{\langle y_i, x_j\rangle : i < j < n+1\} \\ r & := & x_n \\ L & := & \{\langle x_i, y_j\rangle : i \leq j < n\} \cup \{\langle y_i, y_j\rangle : i < j < n\} \\ \mathfrak{C} & := & \{\{y_i, x_{i+1}\} : i < n\} \cup \{\{x_0\}\} \end{array}$$

Proof. Suppose that $\ll = \varnothing$. Suppose first that there are $\mathfrak{x}$ and $\mathfrak{y}$ such that $\mathfrak{x} \not\leq \mathfrak{y}$ and $\mathfrak{y} \not\leq \mathfrak{x}$. Then $\mathfrak{x}$ wc–commands $\mathfrak{y}$ and $\mathfrak{y}$ wc–commands $\mathfrak{x}$. This however contradicts LCA. (For if $\mathfrak{x}$ are $\mathfrak{y}$ are $\leq$–incomparable, so are any of their leaves, and this means that they precede each other.) So, for any two blocks, either $\mathfrak{x} \leq \mathfrak{y}$ or $\mathfrak{y} \leq \mathfrak{x}$. Hence $\mathfrak{C}$ is linearly ordered by $<$. It follows that there can be only one $<$–minimal block, $\mathfrak{x}_0$. It consists of one node, by assumption that no nonbranching nonminimal segment exists. Put $\mathfrak{x}_0 := \{x_0\}$. We have n blocks, $\mathfrak{x}_i$, $i < n$, such that $\mathfrak{x}_i \prec \mathfrak{x}_{i+1}$ and $\mathfrak{x}_i$ is adjoined to $\mathfrak{x}_{i+1}$. (For $\mathfrak{x}_i \ll \mathfrak{x}_{i+1}$ does not hold.) Hence $\mathfrak{x}_{i+1}$ is a two segment block, assuming that there are no nonminimal, nonbranching nodes. Call the lower segment y_i and the upper segment x_{i+1}. Then $y_i < x_{i+1}$, by definition. Furthermore, $x_i < x_{i+1}$. x_i is incomparable with y_i. This is exactly the structure of $\mathfrak{A}$. $\square$

Now we turn to the structure of K(AWC)–structures in general. Here again the structural constraints imposed by LCA are quite drastic. First, we cannot have two adjuncts to the same block unless exactly one of them is a head. For if $\mathfrak{x}_1$ and $\mathfrak{x}_2$ are adjoined to $\mathfrak{y}$, then they exclude each other, and they mutually c–command each other. In order for the leaves of $\mathfrak{x}_1$ to be left of the leaves of $\mathfrak{x}_2$ (in the sense of $\rtimes$) we must have that $\mathfrak{x}_2$ is not a head. For the same reason $\mathfrak{x}_1$ must be a head. Further, if there is an adjunct to a block $\mathfrak{x}$, its leaves must be to the left (in terms of $\rtimes$) of the strong daughters of $\mathfrak{x}$. For the adjunct ac–commands the daughters of $\mathfrak{x}$.

Lemma 39. *Let $\mathfrak{A}$ be a K(AWC)–structure, and $\mathfrak{x}$ and $\mathfrak{y}$ blocks of $\mathfrak{A}$. Suppose that the constituents $\gamma(\mathfrak{x})$ and $\gamma(\mathfrak{y})$ are not morphological heads. Then $\mathfrak{x} \leq \mathfrak{y}$ or $\mathfrak{y} \leq \mathfrak{x}$.*

Proof. Suppose the theorem fails. Then there exist $\mathfrak{x}$ and $\mathfrak{y}$ which head constituents that are not morphological heads but are incomparable with respect to $\leq$. We can actually assume that $\mathfrak{x}$ and $\mathfrak{y}$ are maximal with this property. They are not adjuncts of the same node, however, as we have seen. So, either one is adjoined to the other, or they are both $\ll$–daughters of the same node. If $\mathfrak{x}$ is adjoined to $\mathfrak{y}$, we know that $\mathfrak{x}$ is a head. Likewise, $\mathfrak{y}$ is not adjoined to $\mathfrak{x}$. So, they are $\ll$–daughters of some block $\mathfrak{m}$. There exists $\mathfrak{u}_1, \mathfrak{u}_2 \leq \mathfrak{x}$ such that $\mathfrak{u}_1 \ll \mathfrak{u}_2$, and likewise $\mathfrak{y}$ contains $\mathfrak{v}_1$ and $\mathfrak{v}_2$ such that $\mathfrak{v}_1 \ll \mathfrak{v}_2$. In particular, $\mathfrak{u}_1 \neq \mathfrak{x}$ and $\mathfrak{v}_1 \neq \mathfrak{y}$. Now, $\mathfrak{x}$ ac–commands $\mathfrak{v}_1$. For every block properly including $\mathfrak{x}$ also includes $\mathfrak{m}$ (or is $= \mathfrak{m}$), and so contains $\mathfrak{v}_1$. $\mathfrak{x}$ excludes $\mathfrak{v}_1$, and so $\mathfrak{x}$ wc–commands $\mathfrak{v}_2$. But the $\ll$–mother of $\mathfrak{v}_2$ is $\leq \mathfrak{v}_1$, whence $\mathfrak{v}_2$ does not wc–command $\mathfrak{x}$. This shows that $\mathfrak{x}$ awc–commands $\mathfrak{v}_2$. In the same way we see that $\mathfrak{y}$ awc–commands $\mathfrak{u}_1$, a contradiction to LCA. ☐

Lemma 40. *Let $\mathfrak{A}$ be a K(ASC)–structure, and $\mathfrak{x}$ and $\mathfrak{y}$ distinct blocks of $\mathfrak{A}$. Suppose that the constituents $\gamma(\mathfrak{x})$ and $\gamma(\mathfrak{y})$ are not morphological heads. Then $\mathfrak{x} \ll \mathfrak{y}$ or $\mathfrak{y} \ll \mathfrak{x}$.*

Proof. By the previous theorem, $\mathfrak{x} \geq \mathfrak{y}$ or $\mathfrak{y} \geq \mathfrak{x}$. Assume the first. Suppose that $\mathfrak{x} \gg \mathfrak{y}$ does not hold. Then $\mathfrak{y}$ is adjoined to $\mathfrak{x}$ and must be a head. Contradiction. Likewise, if $\mathfrak{y} \geq \mathfrak{x}$ then also $\mathfrak{y} \gg \mathfrak{x}$, showing the theorem. ☐

It now follows that the relation $\ll$ is rather simple. It is a disjoint sum of right branching structures!

Definition 41. $\mathfrak{A}$ is **strictly complex** if for every leaf $\mathfrak{r}$ there is an $\mathfrak{x}$ such that $\mathfrak{r} \ll \mathfrak{x}$.

It is not hard to deduce the following

Theorem 42. *Let $\mathfrak{A}$ be an adjunction structure. Then $\mathfrak{A}$ is a K(AWC)–structure iff for every block $\mathfrak{x}$ the following holds.*

(1) *At most one block is adjoined to $\mathfrak{x}$. Call it $\mathfrak{s}$. $\mathfrak{s}$ is a morphological head.*

(2) *At most two blocks are immediately included by* $\mathfrak{x}$*. If there are two, one is a morphological head (we call it* $\mathfrak{h}$*) the other is not (we call it* $\mathfrak{c}$*).*
(3) $\mathfrak{c}$ *is strictly complex.*

The relative ordering is $\mathfrak{s} \rtimes \mathfrak{h} \rtimes \mathfrak{c}$*.*

Unlike Kayne we deduce also the requirement that $\mathfrak{s}$ is a morphological head. To get the result of Kayne we must shift to the definition which Kayne takes.

Definition 43. Let $\mathfrak{A}$ be an adjunction structure, and $\mathfrak{x}$ and $\mathfrak{y}$ be blocks. $\mathfrak{x}$ **sc–commands** $\mathfrak{y}$ iff $\mathfrak{x} \not\geq \mathfrak{y}$, and for every block $\mathfrak{u}$ such that $\mathfrak{u} \gg \mathfrak{x}$ also $\mathfrak{u} \gg \mathfrak{y}$ or $\mathfrak{u} = \mathfrak{y}$.

This definition is the version of c–command that takes cones and positions to be strong. The non–inclusion condition remains.

Definition 44. $\mathfrak{x}$ **asc–commands** $\mathfrak{y}$ if neither $\mathfrak{x} \gg \mathfrak{y}$ nor $\mathfrak{y} \gg \mathfrak{x}$, and if $\mathfrak{x}$ sc–commands $\mathfrak{y}$ but $\mathfrak{y}$ does not sc–command $\mathfrak{x}$.

This definition defines the same set of trees, since if blocks are all trivial, and if $\mathfrak{x} = \{x\}$ and $\mathfrak{y} = \{y\}$, then $\mathfrak{x}$ sc–commands $\mathfrak{y}$ iff x c–commands y and $y \not\geq x$. Hence $\mathfrak{x}$ asc–commands $\mathfrak{y}$ iff x ac–commands y. However, notice that the node trace of asc–command is not the same as ac–command on the underlying tree! Now, suppose that $\mathfrak{A}$ is a $K(ASC)$–structure. If $\mathfrak{y}$ has two adjuncts, then they exclude each other, so are incomparable with respect to both $\leq$ and $\ll$. It follows that these adjuncts sc–command each other, which is impossible. So, we retain the fact that there is only one adjunct. If a node has two daughters, exactly one of them is a head. Otherwise, if both are heads, they sc–command each other, a contradiction. If both are non–heads, we find that there are leaves $\mathfrak{r}$ and $\mathfrak{y}$ such that $\mathfrak{x} \rtimes \mathfrak{y}$ and $\mathfrak{y} \rtimes \mathfrak{x}$. So, we are left with almost the same characterization, however with a small difference: adjuncts are allowed to be complex.

Theorem 45 (Kayne)**.** *Let* $\mathfrak{A}$ *be an adjunction structure. Then* $\mathfrak{A}$ *is a K(ASC)–structure iff for every block* $\mathfrak{x}$ *the following holds.*

(1) *At most one block is adjoined to* $\mathfrak{x}$*. Call it* $\mathfrak{s}$*.*
(2) *At most two blocks are immediately included by* $\mathfrak{x}$*. If there are two, one is a morphological head (we call it* $\mathfrak{h}$*) the other is not (we call it* $\mathfrak{c}$*).*
(3) $\mathfrak{c}$ *is strictly complex.*

The relative ordering is $\mathfrak{s} \rtimes \mathfrak{h} \rtimes \mathfrak{c}$*.*

Proof. We have already seen that if $\mathfrak{A}$ is a K(ASC)–structure it satisfies (1) and (2). (3) is also easily established. Suppose that $\mathfrak{c}$ is not complex. Then there exists a leaf $\mathfrak{u} \leq \mathfrak{c}$ which is not included by any member of $\mathfrak{c}$. Then let $\mathfrak{u} \leq \mathfrak{x}$. Any block including $\mathfrak{x}$ must also include $\mathfrak{h}$, by assumption. Since $\mathfrak{h}$ is a head, it is also not complex. Hence, take a leaf $\mathfrak{v}$ of $\mathfrak{h}$. Then neither $\mathfrak{v} \rtimes \mathfrak{u}$ nor $\mathfrak{u} \rtimes \mathfrak{v}$ holds. Now assume conversely that (1), (2) and (3) hold. Let $\mathfrak{u}$ and $\mathfrak{v}$ be distinct leaves. We must show that (a) $\mathfrak{u} \rtimes \mathfrak{v}$ or $\mathfrak{v} \rtimes \mathfrak{u}$, and (b) not both $\mathfrak{u} \rtimes \mathfrak{v}$ and $\mathfrak{v} \rtimes \mathfrak{u}$. Consider the set $U := \{\mathfrak{x} : \mathfrak{x} \geq \mathfrak{u}\}$ and $V := \{\mathfrak{y} : \mathfrak{y} \geq \mathfrak{v}\}$. Clearly, $U \cap V \neq \varnothing$. Let $\mathfrak{b}$ be the minimum of $U \cap V$. Case 1. $\mathfrak{u} = \mathfrak{b}$ or $\mathfrak{v} = \mathfrak{b}$. Without loss of generality $\mathfrak{v} = \mathfrak{b}$. Then $\mathfrak{v}$ does not sc–command $\mathfrak{u}$, but $\mathfrak{u}$ sc–commands $\mathfrak{v}$. So, $\mathfrak{u} \rtimes \mathfrak{v}$. Hence (a) holds. It is clear that (b) also holds. Case 2. $\mathfrak{u} \neq \mathfrak{b}$ and $\mathfrak{v} \neq \mathfrak{b}$. Then let $\mathfrak{c}$ and $\mathfrak{d}$ be $\leq$–daughters of $\mathfrak{b}$ such that $\mathfrak{u} \leq \mathfrak{c}$ and $\mathfrak{v} \leq \mathfrak{d}$. Not both are adjuncts by (1), so we may without loss of generality assume that $\mathfrak{d} \ll \mathfrak{b}$. Case 2a. $\mathfrak{c}$ is an adjunct to $\mathfrak{b}$. Then $\mathfrak{c}$ asc–commands $\mathfrak{d}$. Furthermore, no $\mathfrak{y} \leq \mathfrak{d}$ sc–commands any $\mathfrak{x} \leq \mathfrak{c}$. So, $\mathfrak{u} \rtimes \mathfrak{v}$, but $\mathfrak{v} \rtimes \mathfrak{u}$ does not hold. Case 2b. $\mathfrak{c}$ is not an adjunct to $\mathfrak{b}$. Then both $\mathfrak{c}$ and $\mathfrak{d}$ are $\ll$–daughters, and so by (2) one of them is a head. Assume without loss of generality that $\mathfrak{c}$ is the head. Then $\mathfrak{d}$ is not a head, again by (2). Furthermore, by (3), $\mathfrak{d}$ is strictly complex. Thus $\mathfrak{v} \ll \mathfrak{y}$ for some $\mathfrak{y} \leq \mathfrak{d}$. Furthermore, there is a $\mathfrak{z}$, a $\ll$–daughter of $\mathfrak{y}$ and $\mathfrak{v} \leq \mathfrak{z}$. Then $\mathfrak{c}$ asc–commands $\mathfrak{z}$. Hence $\mathfrak{u} \rtimes \mathfrak{v}$. To show (b), we have to show that $\mathfrak{v} \rtimes \mathfrak{u}$ does not obtain. But $\mathfrak{c}$ sc–commands $\mathfrak{d}$, so $\mathfrak{c}$ sc–commands any block $\leq \mathfrak{d}$. Since $\mathfrak{c}$ is a head, any block of $\mathfrak{c}$ sc–commands any block of $\mathfrak{d}$. Therefore, no block of $\mathfrak{d}$ can asc–command any block of $\mathfrak{c}$. This shows that $\mathfrak{v} \rtimes \mathfrak{u}$ does not hold. The proof is complete. □

These are the requirements as can be found in Kayne.

5.3. **Movement Invariance of Domains.** Finally, we want to discuss an important feature of the new, block based definitions of domains, namely their invariance under movement. In fact, invariance holds with respect to more transformations than just movement. The simplest of them is deletion. Let $\mathfrak{S} = \langle S, <, r, L, \mathfrak{C}\rangle$ be an ordered adjunction structure and $Y \subseteq S$, such that $Y = \uparrow Y$. Then

$$\mathfrak{S} \upharpoonright Y := \langle Y, <_Y, r_Y, L_Y, \mathfrak{C}_Y\rangle \ ,$$

where

$$\begin{array}{lcl} <_Y & := & < \cap Y^2 \\ r_Y & := & r \\ L_Y & := & L \cap Y^2 \\ \mathfrak{C}_Y & := & \{\mathfrak{x} \cap Y : \mathfrak{x} \in \mathfrak{C}, \mathfrak{x} \cap Y \neq \varnothing\} \end{array}$$

Finally, for $\mathfrak{x}$ a block, let $Y := S - \downarrow\mathfrak{x}^\circ \cup \{\mathfrak{x}^\circ\}$. Then $Y = \uparrow Y$ and so $\mathsf{Del}(\mathfrak{S}, \gamma(\mathfrak{x})) := \mathfrak{S} \restriction (S - Y)$ is well–defined and called the result of **deleting** the constituent $\gamma(x)$. If we turn to labelled structures, let $\ell : S \to D$ be a labelling function. Then $\ell_Y : S - Y \to D$ is simply $\ell_Y(x) := \ell(x)$.

The converse operation is called *tagging*. Here we take to adjunction structures and tag the root one structure to some designated node of the first structure. To discuss this properly we need a construction that makes two copies of adjunction structures disjoint in their elements. Let $\mathfrak{S} = \langle S, <, r, L, \mathfrak{C}\rangle$ be an ordered adjunction structure. Put $S^i := S \times \{i\}$, let $h : S \to S^i : x \mapsto \langle x, i\rangle (=: x^i)$. Then h induces a (uniquely) defined adjunction structure, as discussed in Section 2.1. Namely,

$$\begin{array}{rcl} S^i & := & \{x^i : x \in S\} \\ <^i & := & \{\langle x^i, y^i\rangle : x < y\} \\ L^i & := & \{\langle x^i, y^i\rangle : x \, L \, y\} \\ \mathfrak{C}^i & := & \{\{x^i : x \in \mathfrak{c}\} : \mathfrak{c} \in \mathfrak{C}\} \\ \mathfrak{S}^i & := & \langle S^i, <^i, r^i, L^i, \mathfrak{C}^i\rangle \end{array}$$

Now let $\mathfrak{S} = \langle S, <_S, r_S, L_S, \mathfrak{C}_S\rangle$ and $\mathfrak{T} = \langle T, <_T, r_T, L_T, \mathfrak{C}_T\rangle$ be adjunction structures, and $w \in S$ a node. Then

$$\mathsf{Tag}(\mathfrak{S}, w, \mathfrak{T}) := \langle U, <_U, r_U, L_U, \mathfrak{C}_U\rangle ,$$

where

$$\begin{array}{rcl} U & := & S^1 \cup T^2 - \{r_T^2\} \\ <_U & := & <_S^1 \cup <_T^2 \cup \{\langle y^2, x^1\rangle : x \in S, y \in T, w \leq x\} \\ r_U & := & r_S^1 \\ L_U & := & L_S^1 \cup L_T^2 \\ & & \quad \cup\{\langle x^1, y^2\rangle : x \in S, y \in T, x \, L \, w\} \\ & & \quad \cup\{\langle y^2, x^1\rangle : x \in S, y \in T, w \, L \, x\} \\ \mathfrak{C}_U & := & \{\mathfrak{x}^1 : \mathfrak{x} \in \mathfrak{C}_S, w \notin \mathfrak{x}\} \\ & & \quad \cup\{\mathfrak{y}^2 : \mathfrak{x} \in \mathfrak{C}_T, r_T \notin \mathfrak{x}\} \\ & & \quad \cup\{\mathfrak{x}^1 \cup \mathfrak{y}^2 : \mathfrak{x} \in \mathfrak{C}_S, \mathfrak{y} \in \mathfrak{C}_T, w \in \mathfrak{x}, r_T \in \mathfrak{y}\} \end{array}$$

If $\ell_S : S \to D$ and $\ell_T : T \to D$ are labelling functions, the operation of tagging is defined only if $\ell_T(r_T) = \ell_S(w)$. In that case, however, we may define $\ell_U : U \to D$ as follows.

$$\ell_U(x^1) := \ell_S(x), \qquad \ell_U(y^2) := \ell_T(y)$$

Notice that this is exactly as in the case of trees, where we tag the second tree onto the first at node w. The only thing to be remembered is that the block containing the node at which the trees are fused is the join of the blocks containing the counterpart of that node in $\mathfrak{S}$ and $\mathfrak{T}$.

Finally, $\mathsf{Fus}(\mathfrak{S}, w, \mathfrak{T})$, the result of adjoining $\mathfrak{T}$ to $\mathfrak{S}$ at w, is defined as $\langle U, <_U, r_U, L_U, \mathfrak{C}_U\rangle$ where

$$\begin{array}{lcl}
U & := & S^1 \cup T^2 \cup \{w^3\} \\
<_U & := & <^1_S \cup <^2_T \\
 & & \cup\{\langle y^2, x^1\rangle : x \in S, y \in T, w < x\} \\
 & & \cup\{\langle y^2, w^3\rangle : y \in T\} \\
 & & \cup\{\langle x^1, w^3\rangle : x \in S, x \leq w\} \\
 & & \cup\{\langle w^3, x^1\rangle : x \in S, w < x\} \\
r_U & := & \begin{cases} r^2_S & \text{if } w < r_S \\ w^3 & \text{if } w = r_S \end{cases} \\
L_U & := & L^1_S \cup L^2_T \cup \\
 & & \cup\{\langle x^1, y^2\rangle : x \in S, y \in T, x\, L\, w\} \\
 & & \cup\{\langle y^2, x^1\rangle : x \in S, y \in T, w\, L\, w\} \\
 & & \cup\{\langle w^3, x^1\rangle : x \in S, w\, L\, x\} \\
 & & \cup\{\langle x^1, w^3\rangle : x \in S, x\, L\, w\} \\
\mathfrak{C}_U & := & \{\mathfrak{y}^2 : \mathfrak{y} \in \mathfrak{C}_T\} \\
 & & \cup\{\mathfrak{x}^1 : \mathfrak{x} \in \mathfrak{C}_S, w \notin \mathfrak{x}\} \\
 & & \cup\{\mathfrak{x}^1 \cup \{w^3\} : \mathfrak{x} \in \mathfrak{C}_S, w \in \mathfrak{x}\}
\end{array}$$

We call this operation **fusion**, to distinguish it from standard adjunction. If $\ell_S : S \to D$ and $\ell_T : T \to D$ are labelling functions, put $\ell_U : U \to D$ as follows.

$$\ell_U(x^1) := \ell_S(x), \qquad \ell_U(y^2) := \ell_T(y)$$

Clearly, from these elementary operations we can define the standard operations of substitution and adjunction of a constituent of $\mathfrak{S}$ at some node. First, we copy the constituent and tag or fuse it to w. After that we delete the previous copy. Notice that we keep track of the different copies by means of superscripts. If we take the constituent $\gamma(\mathfrak{y})$ and tag or fuse it, then $\gamma(\mathfrak{y})$ splits into $\gamma(\mathfrak{y})^1$ and $\gamma(\mathfrak{y})^2$, which are the old and new copy, respectively.

$$\begin{array}{lcl}
\mathsf{Sub}(\mathfrak{S}, w, \gamma(\mathfrak{y})) & := & \mathsf{Del}(\mathsf{Tag}(\mathfrak{S}, w, \gamma(\mathfrak{y})), \gamma(\mathfrak{y})^1) \\
\mathsf{Adj}(\mathfrak{S}, w, \gamma(\mathfrak{y})) & := & \mathsf{Del}(\mathsf{Fus}(\mathfrak{S}, w, \gamma(\mathfrak{y})), \gamma(\mathfrak{y})^1)
\end{array}$$

With these definitions we can now state and prove the fact that definable command relations (block based) are invariant under these transformations. We call a **schema of binary relations** (or simply **schema**) over labelled, ordered adjunction structures a function X that assigns to each finite labelled ordered adjunction structure $\mathfrak{S}$ a relation $X(\mathfrak{S}) \subseteq S \times S$. Clearly, tight generated command relations are defined as relation schemes.

Definition 46. Let X be a schema of binary relations over labelled, ordered adjunction structures. X is **invariant under deletion** if for any adjunction structure $\mathfrak{S}$, a constituent $\gamma(\mathfrak{y})$,

$$X(\mathsf{Del}(\mathfrak{S},\gamma)) = X(\mathfrak{S}) \cap ((S-Y)\times(S-Y))$$

where $Y := (S - \downarrow\mathfrak{x}^\circ) \cup \{\mathfrak{x}^\circ\}$. X is **invariant under tagging** (**fusion**) if for given structures $\mathfrak{S}$, $\mathfrak{T}$, and $\mathfrak{U} := \mathsf{Tag}(\mathfrak{S}, w, \mathfrak{T})$ ($\mathfrak{U} := \mathsf{Fus}(\mathfrak{S}, w, \mathfrak{T})$)

$$\begin{array}{rcl} X(\mathfrak{S})^1 & = & X(\mathfrak{U}) \cap (S^1 \times S^1) \\ X(\mathfrak{T})^2 & = & X(\mathfrak{U}) \cap (T^2 \times T^2) \end{array}$$

Theorem 47. *Definable command relations (node based) are invariant under deletion and tagging. Definable command relations (block based) are invariant under deletion, tagging and fusion.*

Proof. We will only show the claim for block based command relations. The claim for node based command relations then follows. It is first of all necessary to verify that the claim needs to be proved only for tight relations. To see this, let $X = Y \circ Z$. Then $X(\mathfrak{S}) = Y(\mathfrak{S}) \circ Z(\mathfrak{S})$. So, what we need to show is that in the case of deletion of $\gamma(\mathfrak{x})$ from $\mathfrak{S}$, for any pair of command relations, $(R\cap Y) \circ (S \cap Y) = (R \circ S) \cap Y$, where $Y = S - \downarrow\mathfrak{x}^\circ \cup \{\mathfrak{x}^\circ\}$. However, this holds precisely because Y is upper closed. Similarly, we need that for any pair $\mathfrak{S}$ and $\mathfrak{T}$ of structures, and $w \in S$, for $\mathfrak{U} := \mathsf{Tag}(\mathfrak{S}, w, \mathfrak{T})$ ($\mathsf{Fus}(\mathfrak{S}, w, \mathfrak{T})$), if P and Q are binary relations over S and R and S command relations over T then

$$\begin{array}{rcl} (P \circ Q)^1 & = & P^1 \circ Q^1 \\ (R \circ S)^2 & = & R^2 \circ S^2 \end{array}$$

This clearly holds. The reader may verify the same holds for $X = Y \cap Z$. Hence we are done if the claim is true for tight relations. This however is straightforward to verify. □

6. Prospect

'Generative' literally means: producing. Thus, we will distinguish between a generative grammar and a 'descriptive' grammar. The difference is that the former produces the structures of the language while the latter just tells us how they look like. Grammatical theories that have been proposed can be categorized into either the generative class or the descriptive class, but mixtures are possible and are in fact the rule rather than the exception. If however we are only interested in the set of structures rather than the actual process of obtaining them, then it becomes hard to attribute any significance to this distinction. GPSG may be viewed as a sophisticated version of a generative theory of language, but it can easily be presented as a descriptive theory.

Transformational grammar is presented as a purely generative theory of language. What is more, the structures are obtained by a doubly layered generative process. First, a context–free grammar generates D–structures and then a transformational process is initiated that generates the S–structure, and subsequently the phonetic and logical form. Since the relationship between the generative and the descriptive account on structure in context free grammar is reasonably clear (see [13], [11] and [12]), the focus of attention in transformational grammar is on the second process. A theory is called *representational* or *monostratal* if it makes no use of a transformational component and *derivational* otherwise. [2] Since transformational grammars can produce any recursively enumerable language representational accounts of language that mirror the insights of transformational theory must be more complex than GPSG. One such theory is HPSG.

A derivational theory can be turned into a representational theory. The property that guarantees this is the condition on *recoverability of deletion*. It says that if a transformation τ transforms S_i into S_{i+1} then given S_{i+1} and τ we can reconstruct S_i. Therefore, given S_{i+1} we can guess S_i. Furthermore, if we memorize the subsequent stages of a generative process, then any generative account — even in transformational grammar — can be turned into a descriptive one. However, this is an uninteresting claim. More interesting is the claim that from the beginning of GB the representations used (S–structure, LF) contain enough information to reformulate any derivational account into a representational one. Two factors can be named. First, the restriction of transformations to *Move–α* means that the surface structure contains all material of the intermediate levels, though in somewhat deranged order. Second, the introduction of traces (and the coindexation of traces with their antecedent) made it possible to reconstruct the movement process up to inessential variations. Koster [9] was one of the first to come up with a theory within GB that was purely descriptive. The discussion surrounding the question whether a derivational or generative theory had something to offer that a descriptive or representational theory could not provide has been fought mainly at a philosophical or methodological level and in effect became more and more academic. The arguments in favour of a derivational theory can be grouped into three kinds. (1) Arguments from simplicity. It is claimed that there exist simpler derivational accounts than there can

[2]There is a slight difference between the notion of a representational theory and the notion of a monostratal theory. But for all intents and purposes this boils down to the same in our discussion.

be representational accounts. The main example for this claim is the Head Movement Constraint (HMC). (2) Arguments from reality. In a certain sense — so it is claimed — derivational theories are closer than descriptive theories to what is going on in the human brain. (3) Arguments from methodology. It is said that derivational theories are explanatorily more adequate than representational theories.

This paper contains the first of two parts in which we aim to show that (1) fails. We aim to show in detail that a derivational theory can at low costs be transformed into a representational one, and the resulting theory is no more complicated (in an intuitive as well as in a formal sense) than the previous one. The argument from the HMC works only if a specific choice is made with respect to the representations. However, if (1) fails, then (2) and (3) will be difficult to maintain. The problem is that at least in the relevant linguistic literature evidence is given almost exclusively using syntactic facts. But if we believe (1), then the difference between these theories is merely a notational one. The derivational aspect can be viewed as merely metaphorical. Ironically, in the Minimalist Program it is just that, see Chomsky [4]. Therefore, if (2) and (3) are to make sense, the facts that should decide between these approaches should be sought elsewhere. We are not in a position to make any well–founded claims, but to the best of our knowledge nothing decisive has been found.

In sum, what needs to be seen is that the derivational component in GB can be replaced by a set of surface filters, and that this replacement does not distort the theory in a significant way. We claim exactly that. Unfortunately, this proof is rather long. Not because it is difficult in a mathematical sense, but because a lot of details need to be fixed before a proper proof in the mathematical sense can be given. For the structures have been complicated greatly in Chomsky [3] with the introduction of the category/segment distinction, and once more in Chomsky [4] with the insistance that syntactic relations should be formulated in terms of relations between chains. In the present paper the step from ordered labelled trees to ordered labelled adjunction structures has been looked at in detail and it is shown how the established terminology in linguistics is adapted to the new structures.

References

1. Mark C. Baker. *Incorporation. A theory of grammatical function changing.* Chicago University Press, 1988.
2. Chris Barker and Geoffrey Pullum. A theory of command relations. *Linguistics and Philosophy*, 13:1–34, 1990.
3. Noam Chomsky. *Barriers.* MIT Press, Cambrigde (Mass.), 1986.

4. Noam Chomsky. A minimalist program for linguistic theory. In K. Hale and Keyser S. J., editors, *The View from Building 20: Essays in Honour of Sylvain Bromberger*, pages 1 – 52. MIT Press, 1993.
5. Noam Chomsky. Bare Phrase Structure. In Gert Webelhuth, editor, *Government and Binding Theory and the Minimalist Program*, pages 385 – 439. Blackwell, 1995.
6. Gisbert Fanselow. *Minimale Syntax.* Number 32 in Groninger Arbeiten zur germanistischen Linguistik. Rijksuniversiteit Groningen, 1991.
7. Hubert Haider. Branching and discharge. Technical Report 23, SFB 340, Universität Stuttgart, 1992.
8. Richard S. Kayne. *The Antisymmetry of Syntax.* Number 25 in Linguistic Inquiry Monographs. MIT Press, 1994.
9. Jan Koster. *Domains and Dynasties: the Radical Autonomy of Syntax.* Foris, Dordrecht, 1986.
10. Marcus Kracht. Mathematical aspects of command relations. In *Proceedings of the EACL 93*, pages 240 – 249, 1993.
11. Marcus Kracht. Syntactic Codes and Grammar Refinement. *Journal of Logic, Language and Information*, pages 359 – 380, 1995.
12. Marcus Kracht. Inessential features. In Christian Retoré, editor, *Proceedings of LACL 96*, Springer Lecture Notes in Compter Science. Springer, 1997.
13. James Rogers. *Studies in the Logic of Trees with Applications to Grammar Formalisms.* PhD thesis, Department of Computer and Information Sciences, University of Delaware, 1994.
14. Wolfgang Sternefeld. *Syntaktische Grenzen. Chomskys Barrierentheorie und ihre Weiterentwicklungen.* Westdeutscher Verlag, Opladen, 1991.

II. Mathematisches Institut, Arnimallee 3, D – 14195 Berlin
E-mail address: kracht@math.fu-berlin.de