How linguistic constraints on movement conspire to yield languages analyzable with a restricted form of LIGs*

Jens Michaelis Christian Wartena

Universität Potsdam
Institut für Linguistik/Innovationskolleg
Postfach 60 15 53
14415 Potsdam
Germany

michael@ling.uni-potsdam.de
wartena@ling.uni-potsdam.de

Abstract. Within a GB–like framework movement out of an object seems to be a valid concept for the description of natural languages whereas movement out of subjects and adjuncts is almost impossible. In a framework in which the notion object is determined configurationally, this condition on movement restricts the class of licensed structures. We will argue that this condition restricts as well the class of languages that can be generated. In order to study formal properties we transfer this condition to linear indexed grammars and define two new formalisms: right linear indexed grammars and extended right linear indexed grammar. The classes of languages generated are proper subclasses of the class of linear indexed languages. In fact, we show that right linear indexed grammars are weakly equivalent to context free grammars, despite of the possibility to have in a derivation for each nonterminal an associated index stack (of unbounded depth).

1 Motivation

Within a GB–like framework movement out of an object seems to be a valid concept for the description of natural languages whereas movement out of subjects, adjuncts and indirect objects is almost impossible (see e.g. Müller

*This paper has been worked out within the *Innovationskolleg 'Formale Modelle kognitiver Komplexität' (INK12)* funded by the Deutsche Forschungsgemeinschaft (DFG). Especially, we want to thank Peter Staudacher for constructive discussions and many helpfull comments.

1995 for a survey). There are a few counterexamples like wh–movement out of infinitival subject sentences in German. Anyhow, there is no possibility to pursue this kind of extraction recursively, i.e. there are no similar constructions in which a constituent is moved out of the subject of a subject. Thus, in a formal grammar one could, if need be, ignore the correct analysis and solve the problem by adding adequate nonterminals and corresponding productions. For the moment we will simply assume the initial observation to be correct and investigate the consequences of this constraint for formal grammar models.

In a framework in which such notions as subject and object are determined configurationally, objects are usually realised as complements. A complement is a constituent whose root is the sister of a head (of a projection). According to the linearisation theory of Kayne (1995) and its adoption by Chomsky (1995), complements universally appear to the right of a head whereas specifiers (the typical position of e.g. subjects) and adjuncts appear to the left. Given that the landing site of a moved element has to be a c–commanding position, the only kind of allowed movement is raising out of a constituent whose root has no sister to its right.[1] As for verb final languages the linearisation theory of Kayne seems to be too restrictive. An alternative theory is constituted by the *Binary Branching Conjecture* proposed by Haider (1993). According to this conjecture specifiers and adjuncts appear left of the head as well. The position of complements is lexically determined by the head, either to the left or to the right. The only kind of movement allowed in this framework, therefore, is raising out of a constituent whose root has no nonterminal sister to its right.

In this paper we will not directly investigate whether the mentioned constraints have any positive consequences for the computational complexity of the syntactical system, but transfer them to a formal system with well known properties. A type of grammar that seems to be very well suited to simulate movement is formed by Linear Indexed Grammars (LIGs) (Gazdar 1988). In an LIG the dependency between two positions created by movement in transformational theories can be accounted for by marking all nodes between the moved element (starting with a sister of this element) and its trace with an index symbol. The function of the index symbols is almost the same as that of the so called slash–features in Generalized Phrase Structure Grammar (GPSG). In contrast to the slash feature, index symbols can be stacked. That is to say, each nonterminal node can have an associated stack of unbounded depth with symbols taken from some finite set. In section 2 we will define Right Linear Indexed Grammars (RLIGs), which arise, if we carry over the condition on movement and Kayne's linearisation theory to

[1]Throughout the paper we will use the following definition of c–command: Given a tree, a node A c–commands a node B, iff A does not dominate B and every node dominating A dominates B.

the LIG–formalism. We show that RLIGs are weakly equivalent to CFGs, but still allow to associate an index stack of unbounded length with each nonterminal node. Subsequently, we present a method to construct a weakly equivalent CFG for a given RLIG. The fourth section deals with some consequences of the less restrictive linearisation of Haider's BBC. In the last part of this paper we address the question to what extent the formal results on indexed grammars carry over to linguistic theories.

2 Right Linear Indexed Grammars

With a linear indexed grammar (LIG) movement can be simulated in such a way that the correct distribution of traces and their antecedents is obtained by the use of indices associated with a nonterminal. First ideas on linear indexed grammars were developed in Gazdar 1988. The class of grammars we are interested in can be obtained by restrictions on LIGs. We will show that this class of grammars is weakly equivalent to the class of context free grammars.

Definition 1 (LIG) A *linear indexed grammar (LIG)* G is a five tuple (N, Σ, I, P, S), where N, Σ and I are finite sets, pairwise disjoint, which respectively denote the set of *nonterminal*, *terminal* and *index symbols*. S denotes a distinguished nonterminal (the *start symbol*). P is a finite set of pairs, called *productions*, of one of the forms:

$$1.\ A \rightarrow \alpha B f[\,]\beta \qquad 2.\ Af \rightarrow \alpha B[\,]\beta \qquad 3.\ A \rightarrow w$$

for some $A, B \in N$, $f \in I \cup \{\epsilon\}$, $\alpha, \beta \in (\Sigma \cup N)^*$, $w \in \Sigma^*$, and $[\,]$ an extra symbol not appearing in $N \cup \Sigma \cup I$.

Let $G = (N, \Sigma, I, P, S)$ be an LIG. A string $\sigma \in (NI^* \cup \Sigma)^*$ is called a *sentential form*. A sentential form σ is said to *directly generate (directly derive)* a sentential form τ, written $\sigma \underset{G}{\Rightarrow} \tau$, if either (1,2) or (3):

$$\begin{array}{llll} (1,2) & \sigma = \gamma A f \zeta \delta\ , & Af \rightarrow \alpha B g[\,]\beta \in P\ , & \tau = \gamma \alpha B g \zeta \beta \delta \\ (3) & \sigma = \gamma A \delta\ , & A \rightarrow w \in P\ , & \tau = \gamma w \delta \end{array}$$

for some $A, B \in N$, $f, g \in I \cup \{\epsilon\}$ with $|fg| \leq 1$, $\alpha, \beta \in (\Sigma \cup N)^*$, $w \in \Sigma^*$, $\zeta \in I^*$, and $\gamma, \delta \in (NI^* \cup \Sigma)^*$.[2] The set NI^* is called the set of *objects* in G. It is helpfull to think of an object as a nonterminal associated with a stack of indices. Thus, we see that in case (1,2) such a stack of indices is passed on from the mother to exactly one daughter,[3] while possibly the topmost index

[2] Note that δ cannot start with an index. So, the production $A \rightarrow w$ can only be applied, if no index appears immediately to the right of A.

[3] The daughter, which the stack has to be passed on, is indicated within a production by the symbol $[\,]$.

symbol f is popped from, or a new index g is pushed onto the stack. In this case, with respect to the corresponding occurrences, $Bg\zeta$ is called the *distinguished child* of its mother $Af\zeta$. By $\underset{G}{\Rightarrow}^*$ we denote the reflexive and transitive closure of the relation $\underset{G}{\Rightarrow}$. In the following we will usually omit the subscript G in the corresponding notations, if G is understood. The set $L(G) = \{w \in \Sigma^* \mid S \underset{G}{\Rightarrow}^* w\}$, the language generated by G, is called a *linear indexed language (LIL)*.

For any $n \in \mathbb{N}$ a sequence of sentential forms $d = \alpha_0\alpha_1 \dots \alpha_n$ such that $\alpha_{i-1} \Rightarrow \alpha_i, 1 \leq i \leq n$, is called a *derivation* of α_n from α_0 in G. If in each string α_i the leftmost (rightmost) object is rewritten to directly derive α_{i+1}, then this derivation is called a *leftmost* (rightmost) derivation.

Lemma 2 *Let $G = (N, \Sigma, I, P, S)$ be an LIG. For every $n \in \mathbb{N}$ and every derivation $\alpha_0\alpha_1 \dots \alpha_n$ in G with $\alpha_n \in \Sigma^*$, there is a leftmost (rightmost) derivation of α_n from α_0 in G.* □

Let $G = (N, \Sigma, I, P, S)$ be an LIG. For a given number $n \in \mathbb{N}$ suppose $s = A_0\zeta_0 \dots A_n\zeta_n$ to be some sequence of objects in G and $d = \alpha_0 \dots \alpha_n$ to be some derivation in G with $\alpha_0 = A_0\zeta_0$. Assume each $A_i\zeta_i$ to occur as a particular substring in α_i such that, with respect to these particular occurrences, $A_{i+1}\zeta_{i+1}$ is the distinguished child of $A_i\zeta_i$. Then s is called the *spine* from $A_0\zeta_0$ to $A_n\zeta_n$ *constituted by* d, and $A_n\zeta_n$ is called a *distinguished descendant* of $A_0\zeta_0$ in d. For the sake of convenience, sometimes, we will talk of a *derivation tree T in G*, assuming each node in T to be labelled by an appropriate $X \in NI^* \cup \Sigma \cup \{\epsilon\}$. That is, the labelled structure T corresponds to some derivation $d = \alpha_0\alpha_1 \dots \alpha_n$ in G with $\alpha_0 \in NI^*$. Using the concept of derivation trees with respect to an LIG rather intuitively we will skip further details. It should be clear, what is meant when a path in T is called a spine.

Within a derivation tree corresponding to an LIG–derivation the dependency between the target of a movement and the corresponding base position is respectively coded by a node on which an appropriate index is introduced and by the node (the distinguished descendant) on which this index is absorbed while passing on the stack to the next distinguished child. During the derivation all nodes on the path (the spine) between these two nodes are marked by the corresponding index as well. As noted above, within a range of modern linguistic theories it is assumed that the path between two dependent nodes always coincides with a rightmost branch. Thus, LIGs with only the following type of productions suffice to describe these dependencies:

$$1.\ A \to \alpha Bf[\,] \qquad 2.\ Af \to \alpha B[\,] \qquad 3.\ A \to w$$

for some $A, B \in N$, $f \in I \cup \{\epsilon\}$, $\alpha \in (\Sigma \cup N)^*$, $w \in \Sigma^*$.

Definition 3 (RLIG) An LIG G is called a *right linear indexed grammar (RLIG)*, if the just mentioned restrictions on the productions apply to G.

For RLIGs we will drop the symbol [] in our notation, because it is clear how the indices have to be passed on.

Proposition 4 *The class of RLIGs and that of CFGs are weakly equivalent.*

Proof (sketch).

"$\supseteq$" For every CFG G, there is a RLIG G' such that $L(G) = L(G')$.

Suppose $G = (N, \Sigma, P, S)$ to be a CFG, which w.l.o.g. can be assumed to be in Greibach Normalform. Take $G' = (N, \Sigma, \emptyset, P, S)$ to be the RLIG with $I = \emptyset$. Evidently, G and G' are even strongly equivalent.

"$\subseteq$" For every RLIG G, there is a CFG G' such that $L(G) = L(G')$.

It can be shown that every language derivable by an RLIG $G = (N, \Sigma, P, S)$ can be recognized by a (one way nondeterministic) pushdown automaton ((1N)PDA) and, consequently, is context free. The idea is that an appropiate PDA simulates a leftmost derivation in G. The construction of such a PDA is straightforward, since G can be considered as an "almost" context free phrase structure grammar in the usual sense with nonterminal set $N \cup I$. This is due to the restricted form of the productions in G. Notice that, in terms of strings, applying a production p of the 1st or 3rd form to an object $A\zeta$ is simply a replacing of A by the righthand side of p, a string in $(N \cup I \cup \Sigma)^*$. And in fact, this is even true in case of using a rule of the 2nd form, where the "only" difference consists in the fact that the top symbol of the stack is absorbed. That is to say, the PDA in fact does not need to take care of the inheritance of indices. Only for the application of an index absorbing production a small look–ahead is required.

Let $G = (N, \Sigma, I, P, S)$ be an RLIG. Construct a one way nondeterministic pushdown automaton $M = (Q, \Sigma, (N \cup \Sigma \cup I), \delta, q_0, S, \emptyset)$ for which:

1. $Q = \{q_0\} \cup Q'$, where $Q' = \{q_A | A \in N\}$ with $q_A \neq q_B$ if $A \neq B$ for any $A, B \in N$, and the initial state $q_0 \notin Q'$.
2. δ is a partial mapping from $Q \times (\Sigma \cup \{\epsilon\}) \times (N \cup \Sigma \cup I)$ to finite subsets of $Q \times (NI^* \cup \Sigma)^*$, such that
   $\delta(q_0, a, a) = \{(q_0, \epsilon)\}$ for all $a \in \Sigma$,
   $\delta(q_0, \epsilon, A) = \{(q_0, \alpha) | A \to \alpha \in P\}$ for all $A \in N, \alpha \in (NI^* \cup \Sigma)^*$,
   $\delta(q_0, \epsilon, A) = \{(q_A, \epsilon) | q_A \in Q'\}$ for all $A \in N$,
   $\delta(q_A, \epsilon, f) = \{(q_0, \alpha) | Af \to \alpha \in P\}$ for all $A \in N, f \in I, \alpha \in (N \cup \Sigma)^*$.

In order to establish that M accepts exactly $L(G)$ it can be proved (more generally) that

$$\sigma \underset{G}{\Rightarrow}^* w \text{ iff } (q_0, w, \sigma) \underset{M}{\vdash}^* (q_0, \epsilon, \epsilon) \text{ for each } \sigma \in (NI^* \cup \Sigma)^*,\ w \in \Sigma^*$$

The proof is straightforward. The "only if" follows from an induction on the number of derivation steps in G,[4] the "if" from an induction on the number of recognition steps of the PDA M. The crucial point is that for any index absorbing rule $Af \to \alpha B \in P$, where $A, B \in N$, $f \in I$ and $\alpha \in (N \cup \Sigma)^*$, we have

$$vAf\zeta\delta \underset{G}{\Rightarrow} v\alpha B\zeta\delta \text{ iff } (q_0, w, Af\zeta\delta) \underset{M}{\vdash} (q_A, w, f\zeta\delta) \underset{M}{\vdash} (q_0, w, \alpha B\zeta\delta)$$

for any $v, w \in \Sigma^*$, $\zeta \in I^*$ and $\delta \in (NI^* \cup \Sigma)^*$. $\square$

Note that Proposition 4 provides a generalisation of a result given by Aho (1968) concerning his *right linear indexded right linear grammars (RIRGs)*. In our notation an RIRG is an RLIG in which the righthand side of every rule is of the form aB, aBf or a for some $B \in N$, $a \in \Sigma \cup \{\epsilon\}$ and $f \in I$. Thus, the class of RIRGs is a proper subclass of all RLIGs. But, as shown in Aho 1968 the class of all RIRGs is weakly equivalent to the class of all CFGs, and therefore by Proposition 4 to the class of all RLIGs as well. In fact, to show that every RIRG generates a CFL, for a given RIRG Aho constructs a PDA in the same way as we have done for RLIGs in general. Moreover, Aho points out that also the class of the analogously defined *left linear indexed left linear grammars (LILGs)* is weakly equivalent to the class of CFGs. This will be of some interest in section 4. An LILG is an LIG in which the righthand side of every rule is restricted to the form Ba, $Bf[\,]a$ or a for some $B \in N$, $a \in \Sigma \cup \{\epsilon\}$ and $f \in I$.[5]

3 Transformation to CFGs

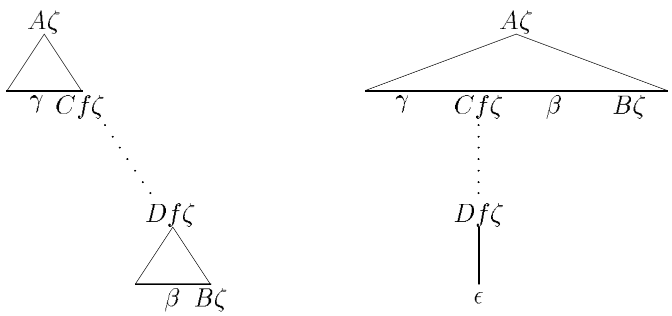
Let $G = (N, \Sigma, I, P, S)$ be an RLIG. Proposition 4 states the existence of a CFG G' which is weakly equivalent to G. Now, we will give a method to directly construct such a CFG $G' = (N', \Sigma, P', S')$: First, the nonterminal set N' is defined as the set of all pairs (A, X) with $A \in N$ and $X \in N \cup \{\epsilon\}$, and the start symbol S' as (S, ϵ). Then the production set P' is defined as[6]

1. $(A, X) \to (\gamma, \epsilon)(C, D)(\beta, \epsilon)(B, X) \in P'$ for all $X \in N \cup \{\epsilon\}$, if $A \to \gamma Cf$, $Df \to \beta B \in P$ for some $A, B, C, D \in N$, $f \in I$ and $\gamma, \beta \in (N \cup \Sigma)^*$.
2. $(A, X) \to (\beta, \epsilon)(B, X) \in P'$ for all $X \in N \cup \{\epsilon\}$, if $A \to \beta B \in P$ for some $A, B \in N$ and $\beta \in (N \cup \Sigma)^*$.

[4] We may consider w.l.o.g. a leftmost derivation (see Lemma 2).

[5] Note that, technically, it is possible to define the concept of *left linear indexed grammars (LLIGs)* analogously to that of RLIGs. Thus, an LLIG is an LIG with only the following type of productions: 1. $A \to Bf[\,]\alpha$, 2. $Af \to B[\,]\alpha$, 3. $A \to w$, with $A, B \in N, f \in I \cup \{\epsilon\}$, $\alpha \in (\Sigma \cup N)^*$, and $w \in \Sigma^*$. Then it is rather easy to see that the results, given so far with respect to RLIGs, hold for LLIGs, too.

[6] For each string $\alpha \in (N \cup \Sigma)^*$ we informally define (α, ϵ) as the string $\alpha' \in (N' \cup \Sigma)^*$, which results from α by replacing every nonterminal $A \in N$ appearing in α by $(A, \epsilon) \in N'$. Thus, for example, $\alpha = abAaBB$ becomes $(\alpha, \epsilon) = ab(A, \epsilon)a(B, \epsilon)(B, \epsilon)$, where $A, B \in N$ and $a, b \in \Sigma$.



(a) a partial RLIG–tree ... (b) ... restructured as a "CFG–tree"

Figure 1: Restructuring of RLIG–trees with respect to spines

3. $(A, \epsilon) \to w \in P'$, if $A \to w \in P$ for some $A \in N$ and $w \in \Sigma^*$.
4. $(A, A) \to \epsilon \in P'$ for all $A \in N$.

Instead of giving a proof that $L(G') = L(G)$, we just briefly outline the motivation for the definition of G'.

We may think of each nonterminal in N' as an object in NI^*. The productions in P', considered as applying to those objects, allow to derive the same strings as G, but with different structures, since G' is context free. Roughly speaking, the productions in P' reflect an algorithm to restructure inductively a derivation tree T_d in G with respect to partial trees, each of which corresponds to a derivation in G which constitutes a spine, namely: a spine from a node $A\zeta$ to a node $B\zeta$, such that an index f is introduced on the distinguished child $Cf\zeta$ of $A\zeta$, and such that this instantiation of f is absorbed when passing the stack from the distinguished descendant $Df\zeta$ to its daughter $B\zeta$ (see Figure 1(a)). Say, $\beta B\zeta$ is the string, derived from $Df\zeta$, while absorbing f. In the restructuring step, the leaves of the local subtree with root $Df\zeta$ are replaced by a single node, labelled ϵ, and become right sisters of $Cf\zeta$ without changing the order of the leaves labelled from left to right by $\beta B\zeta$. That is, $B\zeta$ becomes the rightmost daughter of $A\zeta$ (see Figure 1(b)). The crucial point is that each derivation in an RLIG which constitutes a spine is necessarily rightmost. In terms of derivation trees, the spine coincides with the path from the root to the rightmost leaf. Therefore, in constrast to the general LIG–case, the yield of the initial partial tree is not affected by the restructuring procedure which splits up the spine. Applying productions of P' to nonterminals in N', virtually, such spines are constructed split up already in the mentioned way, directly proceeding from $A\zeta$ with the distinguished descendant $B\zeta$ as rightmost daughter.

Consider a nonterminal $(C, Y) \in N'$. The second component of such a nonterminal provides particular information about the stack η associated with C, appearing as the object $C\eta \in NI^*$. The case $Y = \epsilon$ corresponds to an empty stack (i.e. $\eta = \epsilon$). In accordance with the general LIG–definition only this case allows to rewrite (C,Y) as a string in Σ^* (productions in P' of the 3rd type). The case $Y = D$ for some $D \in N$ corresponds to a non–empty stack (i.e. $\eta \neq \epsilon$). Looking at the definition of P', we see that a second component, other than the empty string, is introduced only if a production p' of the 1st type is applied, say $p' = (A,X) \to (\gamma,\epsilon)(C,D)(\beta,\epsilon)(B,X)$. Then there are productions $p = A \to \gamma C f$ and $q = Df \to \beta B \in P$. In terms of the RLIG G, within p' the index f, introduced by applying p to some object $A\zeta$, not only is suggested to appear subsequently as the topmost stack symbol of a particular distinguished descendant $Df\zeta$ of $Cf\zeta$. But f is also popped off immediately relying on the guess that the production q will be applied to this distinguished descendant. In (C,D) no explicit information must be stored about the stack associated with C. The only information provided is that, at least virtually, there is some topmost stack symbol x, which has to be absorbed by using some production $q_x \in P$ with righthand side Dx. Thus, by the second component of the pair (C,D), implicitly, the information must be stored that rewriting (C,D) in G' has to end up with a string in which the nonterminal (D,D) occurs as the rightmost substring (corresponding to the case that $Df\zeta$ is the distinguished descendant of $Cf\zeta$). Skipping further details we claim that, by induction on the number of derivation steps, it can be shown that

$$A \underset{G}{\Rightarrow}^* wX \text{ (leftmost) iff } (A,X) \underset{G'}{\Rightarrow}^* w$$

for all $A \in N$, $X \in N \cup \{\epsilon\}$ and $w \in \Sigma^*$. Considering the special case $A = S$ and $X = \epsilon$ we see that $L(G) = L(G')$.

4 An Extension

For languages like German and Dutch it seems to be impossible to assume that (at least on surface structure) complements appear to the right of a head. For these languages the linearisation theory proposed by Haider (1993) seems to be better fitted for the linguistic data. Like the proposal of Kayne, Haider's Binary Branching Conjecture (BBC) licenses specifiers and adjuncts only to the left of the main projection. A head may select its complement either to the right or to the left, depending on its category. If we treat a head as a terminal node,[7] movement is resticted to raising out of a constituent whose root is the rightmost *nonterminal* daughter of its mother. This chimes in as well with the movement theory of Müller (1995), in which every NP in a VP constitutes a barrier unless its head can (abstractly) incorporate into the verb. Due to the c–command condition on movement this is only possible if this NP is a sister of the verb. The movement constraints

[7]To be precise, heads are usually treated as preterminals. A preterminal is understood as a nonterminal that can be rewritten as a single terminal only.

at issue applied to a syntactic model obeying the BBC are reflected in the definition of Extended Right Linear Indexed Grammars.

Definition 5 (ERLIG) An LIG G is called an *extended right linear indexed grammar (ERLIG)*, if each of the productions of G is of one of the following forms:

$$1.\ A \to \alpha B f[]w \qquad 2.\ Af \to \alpha B[]w \qquad 3.\ A \to w$$

for some $A, B \in N$, $f \in I \cup \{\epsilon\}$, $\alpha \in (\Sigma \cup N)^*$, $w \in \Sigma^*$.

The generative power of ERLIGs lies beyond that of CFGs, but is in all probability smaller than that of LIGs. With help of the result of Aho (1968) concerning LILGs, it is rather straightforward to show that every language $L = \{a^n b^n c^n L' d^n | n \in \mathbb{N}\}$ is an extended right linear indexed language (ERLIL) if L' is context free. The derivation structure of each sentence in L is basically as sketched in Figure 2. First the outermost substrings of a's and d's are generated and a stack of index symbols is built up. If we reach the point indicated by σ we may push a special symbol onto the stack, which will be popped at the position indicated by τ. On the spine from σ to τ we can use productions of the form $A \to Bf[]x$ and $Af \to B[]x$ (with $A, B \in N$, $f \in I \cup \{\epsilon\}$), $x \in \Sigma \cup \{\epsilon\}$) to derive the substring $x_0 x_1 \dots x_i$. The total amount of indices which are pushed onto the stack and the amount of indices popped off in this part of the derivation has to be equal. Obviously the index stack is passed on correctly from σ to τ. Due to Aho's result, the set of all sequences $x_0 x_1 \dots x_i$ generated between τ and σ is context free and every CFL can be generated as such a set. Considering the fact that only rules of the three forms mentioned before can be used to generate L' because otherwise the stack constructed up to σ would not reach τ, and the fact that the structure in Figure 2 is up to homomorphism the only one for L, it is reasonable to assume that L is only an ERLIL if L' is context free. Thus the linear indexed language $L = \{a^n b^n c^n e^k f^k g^k d^n | n \in N\}$ seems to be no ERLIL, suggesting that the inclusion of ERLILs in LILs is proper.

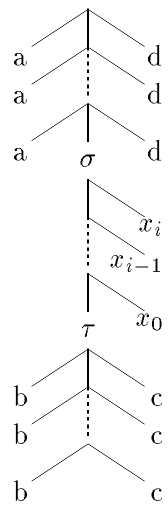


Figure 2: Structure of the language $L = a^n b^n c^n L' d^n$

5 Potentials for the use in linguistic analysis

In the previous sections we have shown that the linguistically rather well motivated restriction on movement to raising out of complements constitutes a real reduction on the generative capacity of LIGs. However, LIGs are only a model for the theories we want to investigate. This model has some inadequacies, of which the most important one is that LIGs can describe nested dependencies only, whereas many linguistic theories use crossing paths as well.

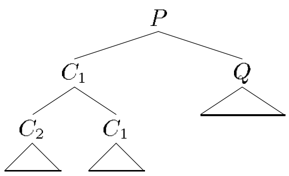


Figure 3: Adjunction Structure

In order to decide whether the conditions on movement discussed above restrict the power of linguistic theories we have to investigate more sophisticated models. However, (E)RLIGs, surprisingly, already constitute a rather good model for some theories that postulate (implicitely or explicitely) an upper bound on the number of constituents that can be moved out of a phrase (e.g. Manzini 1991 and Rizzi 1990). If the existence of such an upper bound is assumed the total number of positions outside of a phrase related to a corresponding base position inside can be unbounded only in case a constituent is moved that contains other moved constituents but not their base positions. A category C_2 can raise along with a category C_1 independent of its trace only if it is *adjoined* to C_1: If C_1 would dominate C_2 it would by definition dominate every position c–commanded by C_2 as well, including the trace of C_2. Thus, raising of C_1 out of a phrase P would create only one dependency between positions inside and outside of P: between C_1 and the trace of C_1. In most transformational theories C_2 can, however, c–command a position that is not dominated by C_1 if it is adjoined to C_1, as is shown in Figure 3. Technically, this is achieved by splitting up categories hosting adjuncts into *segments*. The two nodes labelled C_1 are two segments of the same category. For trees to which this distinction between categories and segments applies c–command is defined in the following way (see Chomsky 1995, Kayne 1995): A c–commands B iff every category dominating A (i.e. all segments of this category dominate A) dominates B and A excludes B (i.e. no segment of A dominates B). Accordingly, C_2 c–commands all nodes dominated by Q and can have a trace that remains within P if C_1 and C_2 are moved.

Furthermore C_2 c–commands C_1 because P is the first category dominating C_1. On the other hand, C_1 does not c–command C_2, because C_1 does not exclude C_2. Consequently, C_2 asymmetrically c–commands C_1 and has to precede C_1 if Kayne's Linear Correspondence Axiom is respected. Thus, recursive application of the considered type of movement and adjunction creates the kind of structure that is exemplified in Figure 4(a) by head movement. We may assume that there is a 1-1–mapping from the class of this kind of structures onto a class of simpler structures, exemplified in Figure 4(b), that make no use of the segment–category distinction. The structures of the latter kind can be described with the RLIG–formalism.

All dependencies created by movement of single categories can be accounted for by adding new nonterminals and productions to the phrase

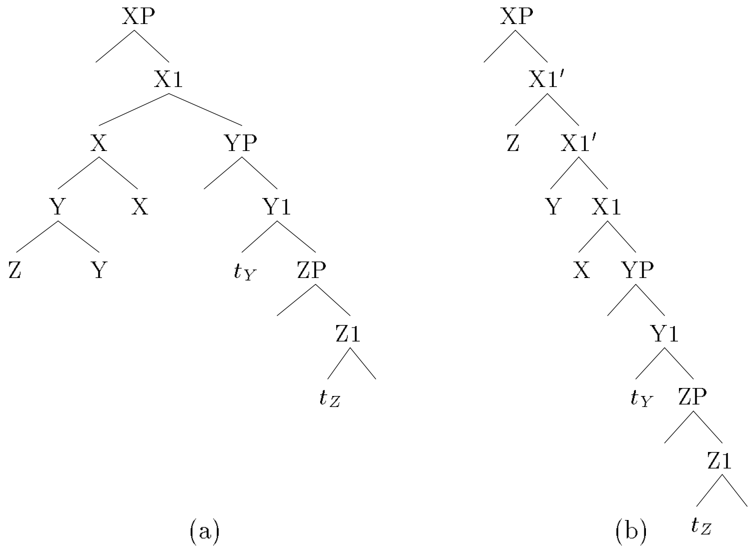


Figure 4: Successive movement and adjunction.

structure grammar or, pictorially speaking, by extending the names of the phrases in between with a slash feature. If there are never two distinct complex categories extracted from the same phrase such that their extraction paths cross, the index stack of the (E)RLIG formalism can be used to code the internal structure of these moving clusters in fact using structures like the one in Figure 4(b). Most syntactic models within the transformational tradition extensively use this kind of movement and adjunction for head movement only. However it is proposed for other types of movement as well. Rudin (1988) e.g. proposes to handle wh–movement in Bulgarian this way[8].

With the ERLIG–formalism, in addition, structures can be handled similarly, in which a constituent moves to the right and adjoins to the righthand side of a higher category. In this way we can e.g. describe the standard structures assumed for cross serial dependencies in Dutch (Evers 1975), in which verbs are raised cyclically to the next verb at the right, in terms of an ERLIG.

6 Conclusion

In this paper we have shown that almost all of the expressive power of LIGs that is needed for the description of natural languages can be preserved if a constraint is added to the formalism so as to obtain a context free system.

[8] Rudin argues that wh–phrases adjoin to the right of their host. Thus the wh–phrases appear in the same order as their traces. Of course, this kind of dependency cannot be described with RLIGs due to inadequacy of the initial LIG–model for natural languages.

This has much in common with e.g. the work of Schabes and Waters (1995) on TAGs. Further this paper gives an example how linguistic principles can constrain the generative capacity of formal grammars.

References

Aho, Alfred V. 1968. Indexed grammars – an extension of context–free grammars. *Journal of the Association for Computing Machinery*, 15:647–671.

Chomsky, Noam. 1995. *The Minimalist Program.* MIT Press, Cambridge (Mass.).

Evers, Arnold. 1975. *The Transformational Cycle in Dutch and German.* Ph.D. thesis, Rijksuniversiteit Utrecht. Reprinted by Indiana University Linguistics Club, Bloomington.

Gazdar, Gerald. 1988. Applicability of indexed grammars to natural languages. In: U. Reyle and C. Rohrer (editors), *Natural Language Parsing and Linguistic Theories*, pp. 69–94. D. Reidel Publishing Company, Dordrecht.

Haider, Hubert. 1993. *Deutsche Syntax Generativ, Vorstudien zur Theorie einer Projectiven Grammatik.* Gunter Narr Verlag, Tübingen.

Kayne, Richard S. 1995. *The Antisymmetry of Syntax*, volume 25 of *Linguisic Inquiry Monograph.* MIT-Press, Cambridge (Mass.).

Manzini, Maria Rita. 1991. *Locality, A theory and Some of Its Empirical Consequences.* MIT-Press, Cambridge (Mass.).

Müller, Gereon. 1995. *A-Bar-Syntax, A Study in Movement Types*, volume 42 of *Studies in Generative Grammar.* Mouton de Gruyter, Berlin.

Rizzi, Luigi. 1990. *Relativized Minimality.* MIT Press, Cambridge (Mass.).

Rudin, Catherine. 1988. On multiple question and multiple wh–fronting. *Natural Language and Linguistic Theory*, 6:445–501.

Schabes, Yves and Richard C. Waters. 1995. Tree insertion grammar: A cubic–time parsable formalism that lexicalizes context–free grammar without changing the trees produced. *Computational Linguistics*, 21:479–513.