Arbeitspapiere des SFB 340
Bericht Nr. 113, Juni 1997

MACROS FOR MINIMALISM: TOWARDS WEAK DESCRIPTIONS OF STRONG STRUCTURES

HANS-PETER KOLB

1. INTRODUCTION

To Rogers (1994) we owe the insight that monadic second order predicate logic with multiple successor functions (MSO, SωS) is in many respects very well suited as a realistic formal base for syntactic theorizing. In particular, his careful formalization of a GB-based grammar of a large fragment of English shows, that a structure-oriented formalism like his MSO-variant $L^2_{K,P}$ is able to adequately express many of the more interesting structural generalizations of the *Principles & Parameters*-approach to natural language syntax in an unprecedentedly transparent fashion. Moreover, the formalism is restrictive: some of Rogers's non-definability results, such as the ones on *free indexing* or the *copy theory of movement* seem to decide longstanding and fierce debates within the *P&P*-literature on purely formal grounds.

However, as the investigations into the descriptive complexity of MSO by Doner (1970), Rabin (1969), Thatcher and Wright (1968) revealed, the agreeable formal properties of this logic, such as its decidability or its ability to define the finiteness of sets, come at the cost of a certain weakness of expressive power: MSO is equivalent with the class of regular tree automata/grammars, and, thereby, with the class of context-free languages. To be more explicit: under its natural interpretation each set of MSO formulæ denotes a *recognizable* set of trees (and *vice versa*). The string language of a recognizable set of trees, however, is always context-free. Thus, MSO-based syntax is strictly context-free. The existence of constructions like the Swiss-German or Dutch verbal complex, which induce non-context-free string languages and non-context-free structures (i.e., non-recognizable sets of trees), respectively (cf. Huybregts 1976,

---

To appear in: Kolb, H.-P. & U. Mönnich: *The Mathematics of Syntactic Structure*.

ISSN 0947-6954/97

1984, Shieber 1985), seems to imply that despite its merits, as a formal metatheory for linguistic theory MSO is too weak.

An important step towards a solution of MSO's expressivity problem is Mönnich (1999)'s result. On the background of an algebraically refined Chomsky hierarchy, which allows the definition of several classes of languages—in particular, a whole hierarchy between CF and CS—via regular tree grammars over unambiguously derivable alphabets of varying complexity plus their respective *yield*-functions, he shows that not only some non-context-free string languages can be captured by context-free means in this way, but that this approach can be generalized to the corresponding structures. I.e., non-recognizable sets of structures can—up to homomorphism—be coded context-freely and, thereby, return into the scope of a MSO treatment.

Since the class of languages covered—Fischer (1968)'s OI family of indexed languages—includes all attested instances of non-context-freeness in natural language,[1] there exists an indirect, to be sure, but completely general way to formally describe the natural languages using MSO.

So, MSO is very well suited to express facts about syntactic structure transparently, and it can talk about all the (sets of) structures linguists want to consider, albeit sometimes somewhat indirectly. End of story—or is it?

Well, no actually, and for two reasons.[2] The first has to do with the question of syntactic *derivations*: Since the (intended) models of (sets of) its formulæ are trees, MSO is perfectly well suited for expression of structural relations between the elements of a single syntactic tree. The derivational view of syntactic structure, however, expressing relations between the elements of *sequences* of trees, is beyond the immediate grasp of MSO. MSO-based syntax is representational. The key to the second problem lies in the innocuous phrase *'up to homomorphism'*: The structures linguistic theory posits for non-context-free constructions are not the structures described by the corresponding (set of) MSO formulæ, but rather their homomorphic images in a tree substitution algebra.

The standard derivational solutions for the phenomena we will be concerned with form some of the better factual arguments for a derivational theory of syntax in that they do not only arrive at the complex results via simple discrete

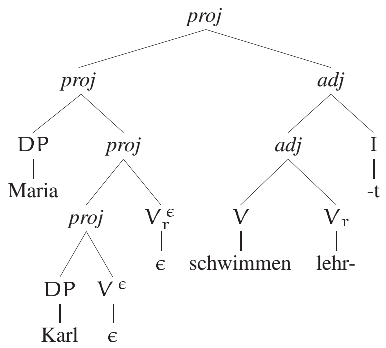
[1] . . . with the one possible, though not particularly probable, exception of some obscure Scandinavian extraction phenomenon which Miller (1991) claims to fall even outside the scope of the indexed languages.

[2] An apparent third one has to do with the fact that Mönnich's construction presupposes the existence of an explicit (macro) tree grammar for the language in question. Neither does linguistic theory provide such a grammar in the general case, nor do there seem to exist general methods of deriving it from the linguistic specifications.

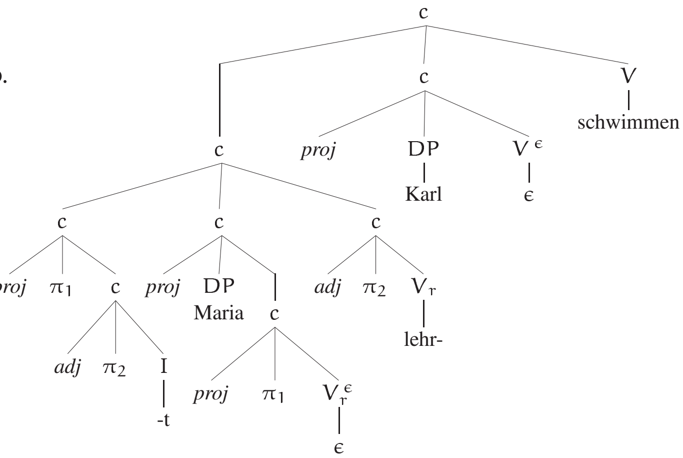
steps but even manage to perspicuously and systematically express the similarities between apparently very different constructions. Therefore the problem of capturing derivations will play a certain rôle in our discussion. Our focus, however, will lie on the second problem, illustrated in (1) using a very simple example:

(1) (. . . dass) Maria Karl schwimmen lehrt
(. . . that) M. K. swim teaches
'(. . . that) Mary teaches Charles to swim'

a.



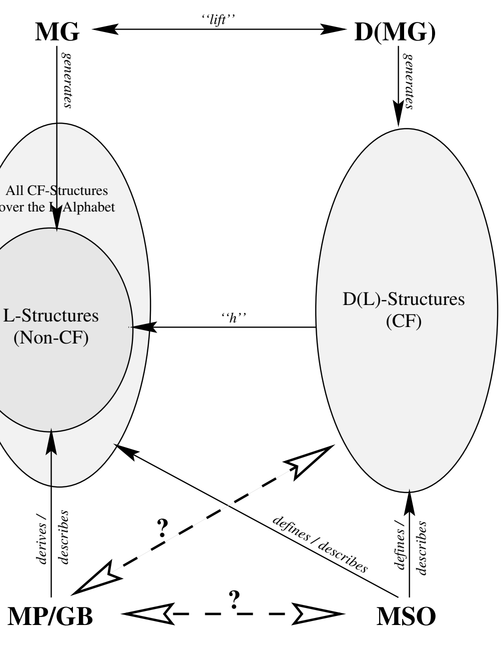
b.



Although it is hard to see in isolation, this little German phrase is an instance of the aforementioned Germanic verbal complex, which gives rise to non-recognizable sets of structures. Modulo notation (which will be properly introduced in section 2), (1a) roughly illustrates the sort of structure it would receive under a generative analysis. In particular, the (*c-command*) relations between the elements of the structure are the ones required by the syntactic theory. This particular structure—as well as any other (tree-)structure over any particular string of words—can be captured using MSO definitions of the elements of the descriptive apparatus of the linguistic theory. But (1a) is also the result of a derivation of some macro tree grammar Γ generating the non-recognizable set of structures $\mathcal{T}(\Gamma)$ of German verb-raising constructions, which MSO is unable to define.

(1b), on the other hand, is a member of $\mathcal{T}(\Gamma^{\mathsf{L}})$, the (recognizable) set of structures admitted by the regular coding, or LIFTing, of Γ, Γ^{L}, which is within the reach of MSO. It is easily seen that it can be unambiguously transformed into (1a), and, indeed, (1a) is (1b)'s unique homomorphic image in the tree substitution algebra. Nevertheless the structure looks very different: Not only does it contain a number of symbols which don't seem to correspond to anything in the linguistic theory, even worse, very few of the structural relations of (1a) are directly represented in (1b), most of them are not only different, but even counter-intuitive, e.g., when the embedded VP seems to c-command the matrix clause.

In a nutshell: While, ideally, MSO can be used transparently to describe any structure syntactic theory might want to assign to some natural language string, there is no set of MSO formulæ—no MSO *theory—over the vocabulary of the corresponding linguistic theory*, which denotes *all and only* the structures/strings of, at least, some natural language.



The use of MSO as a formal meta theory for syntax can be successful only if the following correspondences can be established in a rigorous and effective way:

A. a direct relation between the class of linguistic theories (in the logical sense) and their MSO counterparts over the derived alphabet;

B. a direct relation between a linguistic “grammar” and the set of structures generated by a LIFTed equivalent tree grammar.

(A) would allow us to “forget” the additional structure of the explicit trees when talking (in MSO) about the linguistic structural relations. It amounts to redefining the linguistic primitives (in MSO) systematically in such a way that, for instance, *Maria* c-commands *Karl* both in (1a) and (b). This leads to a decoupling of the “geometric” and the “linguistic” structural relations reminiscent of the techniques underlying Chomsky (1985)’s distinction of segments and categories (cf. Kolb 1995, for some discussion). (B), on the other hand, involves direct interpretation *in the linguistic theory* of the operative function symbols of the macro tree grammars, and of the composition and projection symbols of their regular encodings. We could then state in MSO facts not just about the linguistic structures, but also about the way they are arrived at, thereby obtaining the means for a formal investigation of the resources of modern generative grammar itself.[3]

While a full formal treatment of these options is beyond its scope this paper will explore their feasibility and identify their respective problems, using a simplified minimalist analysis of the Germanic verb complex as the starting point and primary illustrative example. After a brief discussion of the kind of structures generative grammar posits for this kind of phenomenon and some indication (with some hand waving) of how they could be derived in a minimalist setting, we will construct, in section 2, a macro tree grammar Γ_μ generating the same structures, $\mathcal{T}(\Gamma_\mu)$. Applying Mönnich (1999)’s construction we arrive at its regular derivative Γ_μ^L admitting the recognizable set $\mathcal{T}(\Gamma_\mu^L)$ of explicitly coded structures which, as Mönnich (1999) has proven, has the original structures $\mathcal{T}(\Gamma_\mu)$ as its unique homomorphic image in the tree substitution algebra.

Section 3 takes the central notion of *dominance* as the starting point of an investigation of (A), while (B) is the topic of section 4.

2. VERB RAISING: TWO ROUTES AND A DETOUR

As mentioned in the introduction, this exercise in formal coding is made necessary by the fact, that natural language sports at least some constructions which lead to (i) non-CF string languages, or (ii) to non-CF, i.e., non-recognizable structures, even though the resulting string languages are formally context-free.

[3]This would also dissipate the dilemma alluded to in footnote 2.

Both phenomena show up in the west-germanic languages: the verbal complex of Züritüütsch is an example of (i), while (ii) is exhibited—for different reasons—by the corresponding constructions of Dutch and Standard German (Huybregts 1976, 1984):

(2) a. *(... weil) der Karl die Maria dem Peter den Hans schwimmen lehren helfen lässt* (German: Palindrome language—CF)

b. *(... omdat) Karel Marie Piet Jan laat helpen leren zwemmen* (Dutch: $a^n b^n$—CF)

c. *(... wil) de Karl d'Maria em Peter de Hans laat hälffe lärne schwüme* (Züritüütsch: ww—Non-CF)

'(... because) Charles lets Mary help Peter to teach John to swim'

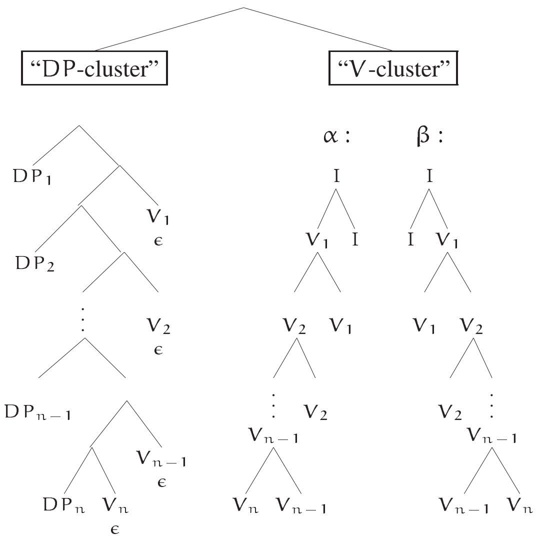
The standard analyses of these constructions involve one property which is problematic from the point of view of context-freeness: In all cases they posit a bipartite structure like the one in figure 1 with basically all DPs on one branch and all the verbs on the other—but with fixed syntactic and semantic relations between the branches, whether visibly marked (as in Züritüütsch, Standard German) or not (Dutch).
As is easily conceived, there is no context-free device which could directly handle the unbounded number of non-local dependencies the structural separation of the two "clusters" enforces. A relatively simple transformational device as the minimalist MERGE-ATTRACT/MOVE machine, however, has little difficulty to combine nearness and separation in the required way. In a nutshell (and suppressing a plethora of important details), this machine is powered by just two operations: MERGE, combining two structures in a rather straightforward way creating a new node by "projecting" one of the input structures and installing both as its sole daughters, and ATTRACT/MOVE, which is dependent on the existence of a pair of attracting/attracted features in the structure so far constructed. It raises an element of the structure carrying the attracted feature and attaches it somewhere higher up in a *check*ing-configuration with the attracting feature, which is deleted in the process. Attracting features come in two flavors, *strong* and *weak*, inducing immediate or delayed movement, respectively. In our case, the moved element will always be adjoined to the element carrying the attracting feature. For details we refer the reader to Chomsky (1995), esp. ch. 4.

The following assumptions suffice for a(n overly simplified) treatment of our example construction, generally known as verb raising (*VR*):

(3) • VR is a lexical property of a certain class of verbs: *lassen-laten-laa* ("let"), *helfen-helpen-hälffe* ("help")... but not *schwimmen-zwemmen-schwüme* ("swim");

a.

b.



c.

	overt CASE	V-cluster
German	+	α
Dutch	−	β
Züritüütsch	+	β

FIGURE 1. The structure of Germanic VR

- VR-verbs take VP-complements;
- VR-verbs (optionally) have a strong V-feature, like the one standardly postulated for the inflectional head I;
- PF serializes complex V_0/I_0 as head-last (German) or head-first (Züritüütsch / Dutch).

Figure 2 depicts a derivation based on these assumptions. $\mathsf{adj}(x, y)$ and $\mathsf{proj}(x, y)$ are shorthands for Chomsky's (1995) $[$*head-of*-y, *head-of*-$y](x, y)$ and *head-of*-$y(x, y)$, respectively. We assume that $\mathsf{adj}(x, y)$ is "spelled-out" xy in German and yx in Dutch and Züritüütsch.

This way the complex result is achieved by a sequence of very simple discrete steps and VR comes as close to being a consequence of formal universals as a basically idiosyncratic phenomenon will ever be able to do.

While the structures derived by the minimalist machine clearly fall outside the limits of a standard context-free (string-)grammar, they can be captured by a *tree*-grammar. These grammars, dating back into the late sixties, differ from the

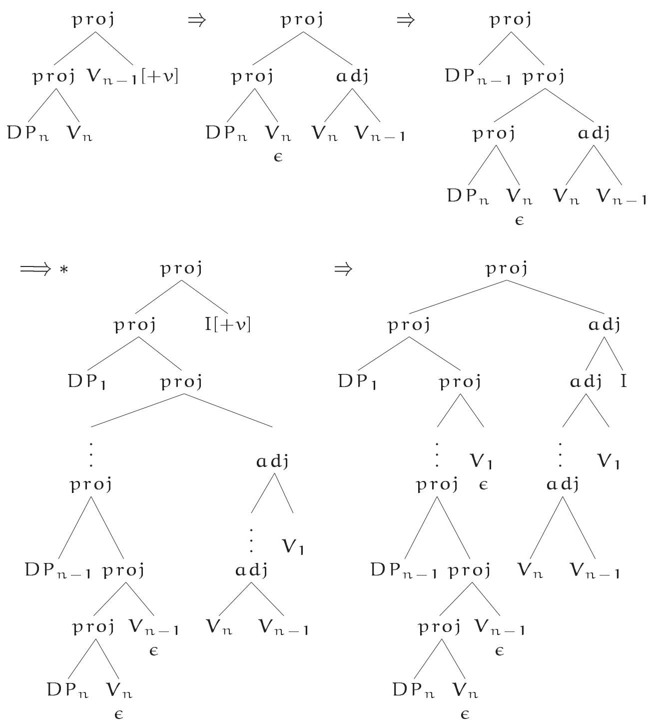


FIGURE 2. A "minimalist" derivation

usual kind in that they "rewrite" trees, not strings. Consequently, their terminals, too, are trees—trees that contain no "rewritable", or *operative*, symbols, that is.

(4)

Definition. A *tree grammar* is a 5-tuple $\langle \Sigma, \mathfrak{F}, S, \mathfrak{V}, \mathfrak{P} \rangle$, where $\Sigma, \mathfrak{F}$ are the ranked alphabets of *inoperatives* and *operatives*, respectively; $S \in \mathfrak{F}$ is the starting symbol; $\mathfrak{V}$ is a countable set of variables; and $\mathfrak{P}$ is a set of productions. The $p \in \mathfrak{P}$ have the form $F(x_i, \cdots, x_n) \longrightarrow t$, with $F \in \mathfrak{F}$, $x_i, \cdots, x_n \in \mathfrak{V}$, and t a term (tree) over $\Sigma \cup \mathfrak{F} \cup \{x_i, \cdots, x_n\}$.

An application of a rule $F(x_i, \cdots, x_n) \longrightarrow t$ "rewrites" a tree rooted in F as the tree t with its respective variables substituted by the daughters of F.

(5) is a trivial example which speaks for itself:

(5) $\Sigma_0 = \{jack, sleeps\}$ $\quad \Sigma_1 = \{VP, DP\}$ $\quad \Sigma_2 = \{S\}$
$\mathfrak{F}_0 = \{\mathsf{S}, \mathsf{F}, \mathsf{G}\}$ $\quad \mathfrak{V} = \emptyset$
$\mathfrak{P} = \{\, \mathsf{S} \to S(\mathsf{F}, \mathsf{G}), \quad \mathsf{F} \to DP(jack), \quad \mathsf{G} \to VP(sleeps)\}$

Tree grammars with $\mathfrak{F}_n = \emptyset, n \neq 0$, are called *regular*. Since they always just substitute some tree for a leaf-node, it is easy to see that they can only generate recognizable sets of trees, *a forteriori* context-free string languages (Mezei and Wright 1967). If $\mathfrak{F}_n, n \neq 0$, is non-empty, that is, if we allow the *operatives* to be parameterized by variables, however, the situation changes. These *context-free* (or *macro-*)tree grammars are capable of generating sets of structures, the *yields* of which belong to the class of mildly context-sensitive languages known as the *indexed* languages. In fact, they characterize this class exactly (Rounds 1970).

As an illustration consider the following macro-tree grammar for the verb-raising construction (2):

(6) Σ_0 (the Lexicon)
$\Sigma_1 = \{\, \mathsf{V}, \mathsf{V}^{\epsilon}, \mathsf{V}_r, \mathsf{V}_r^{\epsilon}, \mathsf{DP}, \mathsf{I}\,\}$ $\quad \Sigma_2 = \{\, \mathsf{proj}, \mathsf{adj}\,\}$
$\mathfrak{F}_0 = \{\, VR \,\}$ $\quad \mathfrak{F}_2 = \{\, +V \,\}$ $\quad \mathsf{S} = VR$ $\quad \mathfrak{V} = \{x, y\}$

$$\mathfrak{P} = \left\{ \begin{array}{rcl} VR & \longrightarrow & +V(\mathsf{proj}(\mathsf{DP}, \mathsf{V}^{\epsilon}), \mathsf{V}) \\ +V(x, y) & \longrightarrow & \mathsf{proj}(x, \mathsf{adj}(y, \mathsf{I})) \\ +V(x, y) & \longrightarrow & +V(\mathsf{proj}(\mathsf{DP}, \mathsf{proj}(x, \mathsf{V}_r^{\epsilon})), \mathsf{adj}(y, \mathsf{V}_r)) \end{array} \right\}$$

plus the obvious rules for Σ_1

"Spell-out" parameterization of adj applies as before.

As the sample derivation in figure 3 illustrates, the minimalist and the tree grammar generated structures are identical.

Any *context-free* tree grammar Γ can be transformed into a *regular* tree grammar Γ^L, which characterizes a (necessarily recognizable) set of trees encoding the instructions necessary to convert them into the ones the original grammar generates (Maibaum 1974). This "LIFTing" is achieved by constructing a new, derived alphabet and translating the terms over the original signature into terms of the derived one via a primitive recursive procedure.

(7)

Definition. For each $n \geq 0$, $\Sigma'_n = \{f' | f \in \Sigma_n\}$ is a new set of symbols; $\mathfrak{V}_k = \{x_i \mid 1 \leq i \leq k\}$; for each $n \geq 1$ and each $i, 1 \geq i \geq n$, π_i^n is a new symbol, the ith projection symbol of sort n; for each $n \geq 0, k \geq 0$

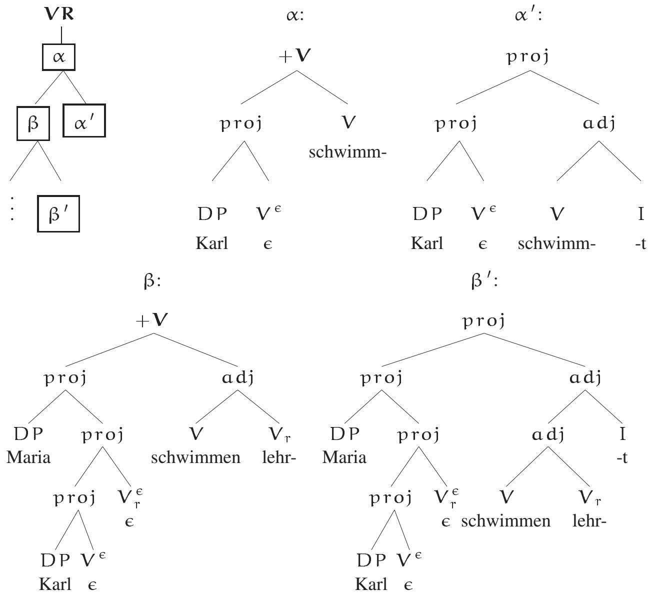


FIGURE 3. A (macro-)tree grammar derivation of VR

the new symbol $c_{n,k}$ is the (n,k)th composition symbol.

$$\Sigma^L_n = \Sigma'_n \cup \{\pi^n_i | 1 \geq i \geq n\} \text{ for } n \geq 1 \qquad \Sigma^L_0 = \Sigma'_0$$

$$\Sigma^L_{n,k} = \{c_{n,k}\} \text{ for } n,k \geq 0 \qquad \Sigma^L_i = \emptyset \text{ otherwise}$$

For $k \geq 0$, $\textsc{lift}^\Sigma_k : T(\Sigma, \mathfrak{X}_k) \to T(\Sigma^L, k)$ is defined as follows:

$$\textsc{lift}^\Sigma_k(x_i) = \pi^k_i \qquad \textsc{lift}^\Sigma_k(f) = c_{0,k}(f') \text{ for } f \in \Sigma_0$$

$$\textsc{lift}^\Sigma_k(f(t_1, \ldots, t_n)) = c_{n,k}(f', \textsc{lift}^\Sigma_k(t_1), \ldots, \textsc{lift}^\Sigma_k(t_n))$$
$$\text{for } f \in \Sigma_n, n \geq 1$$

Note, that this very general procedure allows the translation of any term over the original signature. Obviously, a rule of a grammar Γ is just another term over Σ^Γ, but so is, e.g., any structure *generated* by Γ.

Again simplifying things to the verge of inaccuracy—in particular, omitting all 0- and 1-place composition symbols, (7) yields the following LIFTed grammar for (6):

(8) $\Sigma^L_0 = \Sigma_0 \qquad \Sigma^L_1 = \{ V', V^{\epsilon\prime}, V'_r, V^{\epsilon\prime}_r, DP', I' \}$
$\quad\;\; \Sigma^L_2 = \{ proj', adj', \pi_1, \pi_2 \} \qquad \Sigma^L_{n,k} = \{ c_{n,k} \mid c \in \mathfrak{C} \}$

$$\mathfrak{F}_0 = \{\ VR', +V'\ \} \qquad S = VR' \qquad \mathfrak{V} = \emptyset$$

$$\mathfrak{P} = \left\{ \begin{array}{lcl} VR' & \Longrightarrow & c(+V', c(\mathrm{proj}', DP', V^{\epsilon\prime}), V') \\ +V' & \Longrightarrow & c(\mathrm{proj}', \pi_1, c(\mathrm{adj}', \pi_2, I')) \\ +V' & \Longrightarrow & c(+V', c(\mathrm{proj}', DP', c(\mathrm{proj}', \pi_1, V_r^{\epsilon\prime})), \\ & & \qquad c(\mathrm{adj}', \pi_2, V_r')) \end{array} \right\}$$

plus the rules for Σ_1^L

This grammar leads to structures, given in figure 4, which on first sight don't have much in common with the ones we are really after, i.e., the ones generated by (6).

However, as mentioned before, there is a mapping h from these structures onto structures interpreting the $c \in \mathfrak{C}$ and the $\pi \in \Pi$ the way the names we have given them suggests, *viz.* as compositions and projections, respectively. In fact, h is the *unique* homomorphism into the appropriate tree substitution algebra. Moreover, as Mönnich (1999) has shown, the image of the set of Γ^L structures under this homomorphism are exactly the structures generated by Γ.[4] So in some sense the Γ^L structures, which form a recognizable set and, therefore, can be described by a context-free grammar or, more importantly, by a set of MSO-formulæ, already contain the intended structures. In what follows we will explore how this fact can be brought to bear on a MSO-based metatheoretical treatment of modern generative grammar.

3. Linguistic relations on explicit structures

The minimal requirement for any direct use of the explicitly encoded, recognizable sets of structures is the definition on the derived trees of the linguistic relations that hold in the original structures. The first thing that comes to mind is something like the following:

(9) x $\begin{array}{c}\textit{dominates}^L \ / \ \textit{c-commands}^L \ / \\ \textit{is-in-the-checking-domain-of}^L\end{array}$ y in T^L

if $h(x)$ $\begin{array}{c}\textit{dominates} \ / \ \textit{c-commands} \ / \\ \textit{is-in-the-checking-domain-of}\end{array}$ $h(y)$ in $h(T^L)$.

Actually, such a definition would be valid: from Mönnich (1999) we know that h exists and since any tree can be described by a $L^2_{K,P}$-formula, so can $h(T^L)$. The definitions of *c-commands, dominates* and *is-in-the-checking-domain-of* can either be taken directly from Rogers (1994) or easily derived from his definitions.

[4]For the technical details in their full glory cf. Mönnich (1999). For our purpose it is sufficient to know that this unique mapping exists.

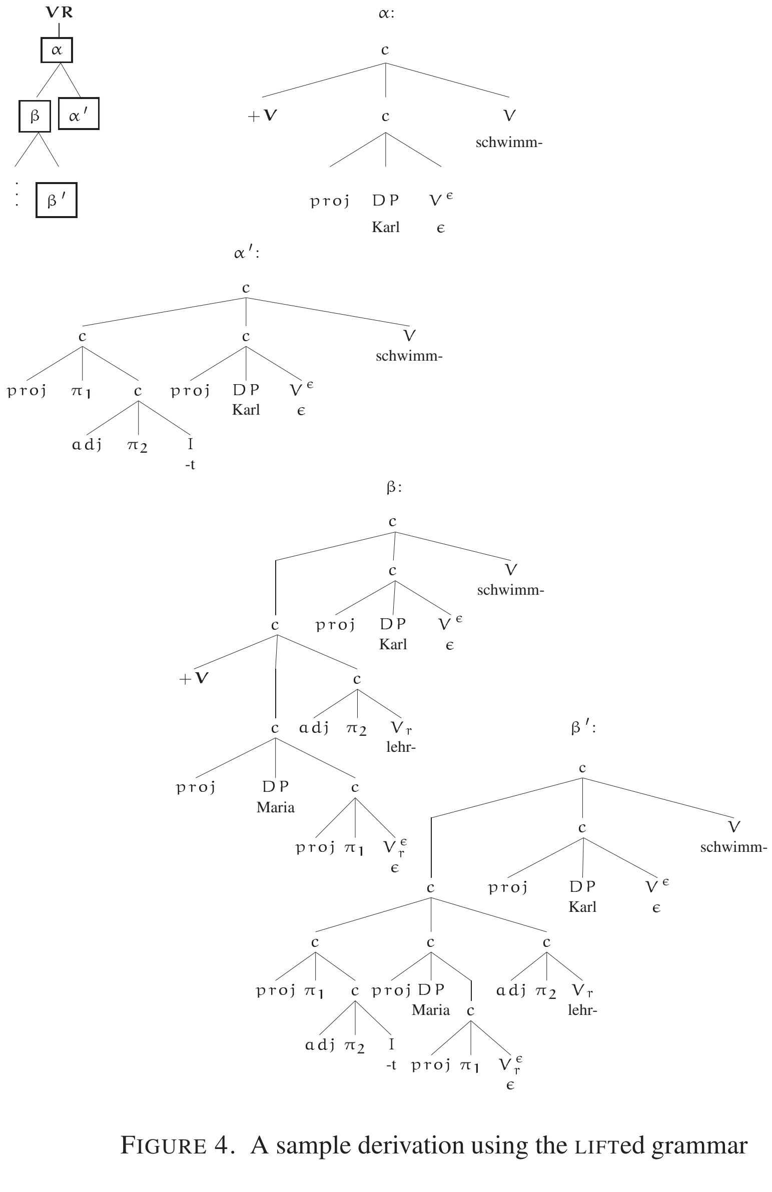


FIGURE 4. A sample derivation using the LIFTed grammar

(9) is not effective, however: There is no way to reconstruct T^L from a set of *dominance*^{L} or *c-command*^{L} statements alone. But even if we ensure the general well-formedness of T^L by other means—definitions à la (9) seem to introduce unnecessary complexity.

A more interesting approach would be to "build h into the definitions." The aim of such an undertaking is to define a set of relations R^h, holding between the nodes $n \in N^L$ of the explicit tree T^L which carry a *"linguistic"* label $l \in \textsc{lift}(\bigcup_{n\geq 0} \Sigma_n)$ in such a way, that when interpreting *dominates*$^h \in R^h$ as a tree order on the set of *"linguistic"* nodes and *precedes*$^h \in R^h$ as the precedence relation on the resulting structure, $\langle\{n \mid n \in N^L \wedge \ell(n) \in \textsc{lift}(\bigcup_{n\geq 0} \Sigma_n)\}$, *dominates*^{h}, *precedes*$^h\rangle$ is in fact $h(T^L)$. What we are after is a *syntactic interpretation* (cf., e.g., Ebbinghaus et al. 1996, :131ff) of T in T^L. This involves finding a set of formulæ Φ which maps the relevant relations R into T^L in such a way that $\Phi^{-1}(T^L) = \langle \varphi_{N^L}^{-1}, \varphi_R^{-1}\rangle$ is the intended, "linguistic" structure corresponding to T^L. Taking (following Rogers 1994) $\{\triangleleft, \triangleleft^\star, \prec\}$ as the primitive structural relations on the intended trees and assuming that the corresponding $\{\blacktriangleleft, \blacktriangleleft^\star, \lessdot\}$ are definable on the $\textsc{lift}$ed structures, Φ is given by $\{\varphi_{N^L} =$ *an appropriate restriction of the nodes of* T^L, $\varphi_\triangleleft = \blacktriangleleft$, $\varphi_{\triangleleft^\star} = \blacktriangleleft^\star$, $\varphi_\prec = \lessdot\}$, and $T = \Phi^{-1}(T^L) = \langle \varphi_N^{-1}, \varphi_\triangleleft^{-1}, \varphi_{\triangleleft^\star}^{-1}, \varphi_\prec^{-1}\rangle$ for any explicit tree T^L Again, since h exists, this construction is possible in principle. What complicates matters in our case are two additional requirements: In order to count as a solution for the expressivity problem Φ must be cast in MSO, and to pave the way towards an interface to structures generated by linguistic means, Φ should be effective, i.e., Φ applied to some linguistic tree should enable us to derive a(n equivalence class of) T^L-structure(s). As a corollary, this requires that linguistic relations between two nodes can be determined *locally* in T^L, i.e., without reference to the complete tree.

If successful, one would replace $\triangleleft$, $\triangleleft^\star$, $\triangleleft^+$ and $\prec$ in Rogers' *Government/-Binding*-based formalization of English by $\blacktriangleleft$, $\blacktriangleleft^\star$, $\blacktriangleleft^+$ and $\lessdot$, respectively, and receive the same analyses, only now stated on the $\textsc{lift}$ed trees. But in addition the extra structure of these trees allows the (decidable) formulation of conditions, which may lead to (intended) structures beyond any direct $L^2_{K,P}$ treatment.[5]

[5]This last sentence sets off the alarm bells immediately: If some set of structures is not MSO-definable, it should remain undefinable if derived from some other set. However, nothing has been said so far about the nature of these additional conditions. Below it will become clear that such constraints would crucially involve the additional, non-linguistic ingredients of T^L.

On first glance, however, the explicit trees as defined in Mönnich (1999) do not seem to contain enough information to locally determine the result of applying the homomorphism—only top-down evaluation *of the full tree* will eventually lead to the proper configuration. But some examination of the structures involved shows that, at least as long as we stick to terminal structures generatable by some regular tree grammar Γ^L which is the result of LIFTing a context-free (macro-)tree grammar Γ, this problem can be overcome to a fairly large extent, at least as far as dominance and its derivatives are concerned.

Unfortunately, this conclusion does not extend to precedence, however. Even though precedence is trivially represented in any SnS-tree via the ordering of the successor functions, the derived notion of precedence between "linguistic" entities poses additional problems. To see why, consider the grammar with the inoperatives $\Sigma = \{0, 1, \cdot\}$, the operatives $\mathfrak{F} = \{S, F\}$, the variables $\mathfrak{V} = \{x, y\}$ and the following rules:

$$\mathfrak{P} = \left\{ \begin{array}{rcl} S & \to & F(1, 0) \\ F(x, y) & \to & F(\cdot(1, x), \cdot(0, y)) \\ F(x, y) & \to & \cdot(y, x) \end{array} \right\} \qquad (\dagger)$$

This grammar generates binary trees with an equal number of 0s and 1s as the leaves and where all the 0s precede all 1s.[6] The trees generated by the LIFTed grammar, however, do a marvelous job in obfuscating this fact, as figure 5 shows.

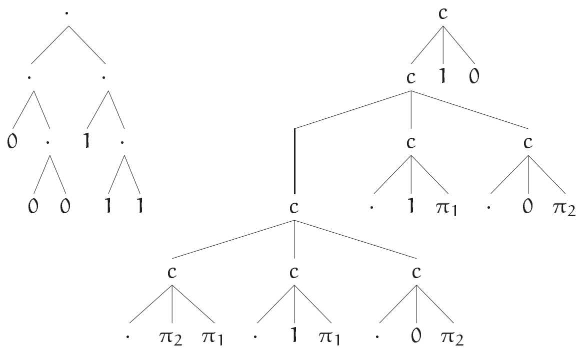


FIGURE 5. Derivation trees for 000111

It is in particular the last rule (†), perfidiously reversing the order of the parameters, which ensures that there is no way to determine the intended precedence

[6]In fact, except for the root, every node dominating some 0 precedes every node not dominating it which dominates some 1.

relation locally, without evaluating the complete tree. And since (†) is a perfectly well-formed context-free (macro-)tree grammar rule, it follows that $\triangleleft$ is not definable in any sensible manner.[7]

So a simple plug-in replacement for Rogers' primitives is out of reach. However, initiated by Kayne (1994), the rôle of precedence as a primitive of syntactic theory has been widely questioned lately—with an array of proposals ranging from its derivation from the hierarchical structure, i.e., from dominance (in Kayne 1994) to its exclusion from syntax proper, treating linear order in the morphophonemic module. Chomsky (1995) assumes a combination of these positions which doesn't use precedence as a syntactic primitive, either. But even in the Government/Binding framework Rogers bases his formalization on, there is little reference to precedence at all. The one big exception is the notion of *directionality of government*. Consequently, Rogers uses precedence mainly in his formalizations of government-related concepts like *head*, *complement*, etc. However, if the notion of *head* is characterized without reference to precedence, precedence should follow from hierarchical structure and the value of the directionality parameter alone, providing yet another example of the ubiquitous decoupling of "geometric" and "linguistic" structural relations. So even without a direct counterpart of $\prec$ the general approach is still worthwhile.

The language we intend to use is Rogers' variant of monadic second order predicate logic, $L^2_{K,P}$, with the following ingredients:

(10)
- three disjoint, countable sets of *individual constants* $\mathbf{K}$, *predicate constants* $\mathbf{P}$, and *ranked variables* $\mathbf{X} = \mathbf{X}^0 \cup \mathbf{X}^1$
- four binary predicates: *equality* $\approx$, plus the tree predicates *parent* $\triangleleft$, *dominance* $\triangleleft^\star$, and *left-of* $\prec$
- the usual connectives, quantifiers, brackets

where the $k \in \mathbf{K}$ are be interpreted as nodes, the $p \in \mathbf{P}$ as properties (or labels) of nodes, and the $x \in \mathbf{X}^0$ and $X \in \mathbf{X}^1$ range over $\mathbf{K}$ and $\mathbf{P}$, respectively. We will write $X(x)$ for 'x is in the set assigned to X'. The syntax is the standard first

[7]Note, that this problem stems from our insistence to talk about structures *generated by some* LIFT*ed grammar*, as opposed to to LIFT*ed structures*: LIFTing the trees generated by some macro tree grammar yields structures without any "projection" symbols. $\triangleleft$ on the terminal nodes of the intended trees is just $\prec$ and the calculation of $\triangleleft$ involving non-terminal nodes, as well as of $\blacktriangleleft$, is vastly simplified. However, this does not solve the recognizability problem: LIFTed trees fully inherit the properties of the underlying ones—in the verb-raising case, for instance, they exhibit just the type of systematic dependencies of unbounded number and depth between unconnected subtrees that made the original set of structures non-recognizable. In contrast, structures generated by the LIFTed grammar are, in a certain sense, encodings of the additional control needed to map effectively recognizable sets of structures onto (potentially) non-recognizable ones.

order predicate logic syntax extended by quantification over monadic predicates: $\forall X \in X^1$, if φ is a formula, so are $\exists X\varphi$ and $\forall X\varphi$.

No extra n-place predicates for $n > 1$ are allowed unless they can be explicitly defined. Monadic predicates, however, are always possible since they can be added freely to $\mathbf{P}$.

The intended models of $L^2_{K,P}$ are tree domains $\mathbb{T}$ under their their natural interpretation $\mathbb{T}^\natural$, where $\triangleleft$, $\triangleleft^\star$, $\prec$ actually have their intuitive meaning, plus interpretations for the additional individual and predicate constants, i.e., just labeled trees (cf. Rogers 1994, for details).

To start we note some general facts about Γ^L-structures, obtained by inspection of the LIFTing operation:

(11) a. Our trees feature three families of labels: the "linguistic" symbols, i.e., the lifted inoperatives of the underlying macro-grammar, $\mathfrak{L} = \text{LIFT}(\bigcup_{n \geq 0} \Sigma_n)$; the "composition" symbols $\mathfrak{C} = \{c_{n,k}\}, n, k \geq 0$; and the "projection" symbols Π. Since we will have to make crucial reference to certain subsets of Π below, we have to be a little less cursory here: just like the $c \in \mathfrak{C}$, each $\pi^k_i \in \Pi$ has some information attached which allows for local well-formedness checking of the structures which sport it as a label (the k), and codes some information on which (projection-) function it will eventually denote (the i). While for the purpose of this paper we take the first information for granted, we need to have access to the second. So we let $\Pi_i = \{\pi^k_i \mid k \geq 1\}$ and $\Pi = \bigcup_{i \geq 1} \Pi_i$. We add to $\mathbf{P}$ predicates denoting these families, i.e., $\mathfrak{L}, \mathfrak{C}, \overline{\Pi}$, as well as their constituent sets, in particular all Π_i.[8]

b. All non-terminal nodes in T^L are labeled by some $c \in \mathfrak{C}$. This is due to the fact that the "composition" symbols are the only non-terminals of a LIFTed grammar. No terminal node is labeled by some c.

c. The terminal nodes in T^L are either labeled by some "linguistic" symbol or by some "projection" symbol $\pi \in \Pi_i$. Because of our restriction to terminal structures the third kind of frontier symbols of a (macro-)tree grammar, the operatives $f \in \mathfrak{F}$, never show up.

d. Any "linguistic" node dominating anything in the intended tree is on some left branch in T^L, i.e., it is the left-most daughter^L of some

[8]A fourth family induced by the LIFTing operation, the operative (or macro-)symbols $\mathfrak{F} = \bigcup_{n \geq 1} \mathfrak{F}^L_n$ is irrelevant here since we only consider *terminal* structures, i.e., trees which contain no operative nodes.

$c \in \mathfrak{C}$. This lies in the nature of composition: $c(x_0, x_1, \cdots, x_n)$ evaluates to $x_0(x_1, \cdots, x_n)$.

e. For any node p labeled with some "projection" symbol $\pi \in \Pi_i$ in T^L there is a unique node n (labeled with some $c \in \mathfrak{C}$ by (11b)) which $\triangleleft^+$-s p and whose i-th sister will eventually evaluate to the value of π. Moreover, n will be the first node $\triangleleft^+$-ing p which is on a left branch. This crucial fact is arrived at by easy induction on the construction of Γ^L from Γ.

Now we seem to be in a position to define the $\blacktriangleleft$-family of relations, starting with some auxiliary definitions.

(12) $$\begin{aligned} \textit{edge}_n(x, y) \equiv \exists x_1, \cdots, x_{n-1}(x \triangleleft_L x_1 \wedge \cdots \wedge x \triangleleft_L x_{n-1} \wedge x \triangleleft_L y \\ \wedge x_1 \prec_L x_2 \wedge \cdots \wedge x_{n-1} \prec_L y \wedge \ \forall w(x \triangleleft_L w \\ \wedge w \not\approx x_1 \wedge \cdots \wedge w \not\approx x_{n-1} \wedge w \not\approx y \ . \rightarrow . \ y \prec_L w)) \end{aligned}$$

$$\textit{leftmost}(x) \equiv \forall y(y \triangleleft_L x \rightarrow \textit{edge}_1(y, x))$$

$\textit{edge}_n$ is a definition scheme used to capture the general case of an n-branching tree, for any $n < \omega$. It can be trivially unfolded for any given n.[9]

The complexity of the definition of $\blacktriangleleft$ is hidden in the *evaluates-to* predicate, which determines the actual "filler" of a T^L-node considered a candidate-dominee in T by (11d). There are three cases to account for:

(1) If the node considered carries a "linguistic" label, it evaluates to itself;
(2) if it has a "composition" label c, it evaluates to whatever its functor—by (11d) its leftmost daughter—evaluates to;
(3) if it carries a "projection" label π_i, it evaluates to whatever the node it "points to"—by (11e) the i^{th} sister of the first $\mathfrak{C}$-node on a left branch dominating it—evaluates to.

Note that cases 2 and 3 are inherently recursive.

The treatment of "projection"-nodes makes crucial use of (11e) and of the construction of Π in (11a). Again, to capture the general case with an unbounded (but finite) number of possible macro-parameters leading to an equally large number of Π_i, we need one disjunct/formula for every i, which again we code as a definition scheme.

It seems safe to conjecture, though, that in structures generated via the LIFTing of a tree grammar closely mimicking the corresponding linguistic derivations of the minimalist kind even this small technical complication will vanish: Where

[9] A different way of capturing the *edge*-type of "variable" relations would make direct use of the underlying SnS tree with its successor-functions $\textit{succ}_i$, $1 \leq i \leq n$. Then the edge-relation is just $\textit{edge}_i(x, \textit{succ}_i(x))$.

macros correspond to minimalist operations in any interesting way, the index on any π will never exceed 2 and the branching will be always binary in the intended trees, reducing the actual "counting" required to a binary choice and the technical machinery needed to naught.

(13) *Immediate dominance:*

$$x \blacktriangleleft y \equiv \mathfrak{L}(x) \land \textit{leftmost}(x)$$

– $\ell(x)$ is a "linguistic" symbol on a left branch

$$\land \exists z, w(z \lhd_L x \land z \lhd_L w \land w \not\approx x \land \textit{evaluates-to}(w, y))$$

– with some sister evaluating to a "linguistic" label

$$\begin{aligned}\textit{evaluates-to}(x, y) \equiv\ & \mathfrak{L}(x) \land x \approx y \\ & .\lor .\ \mathfrak{C}(x) \land \exists z(\textit{edge}_1(x, z) \land \textit{evaluates-to}(z, y)) \\ & .\lor .\ \Pi_i(x) \land \exists u, v, z(u \lhd_L^+ x \land \textit{edge}_1(v, z) \land \\ & \qquad \forall w(w \lhd_L^+ x \land u \lhd_L^+ w\ .\rightarrow .\ \neg \textit{leftmost}(w)) \\ & \qquad \land \textit{edge}_{i+1}(v, z) \land \textit{evaluates-to}(z, y))\end{aligned}$$

Unfortunately, (13), as it stands, is not a valid MSO definition.

3.1. **Ensuring definability.** In general, recursive definitions in MSO may lead to undecidability and are therefore disallowed. Fortunately, there are certain techniques to ensure that some relation R which would most naturally be defined recursively has a valid MSO definition. Still, special care has to be taken to establish that the relations defined are well-behaved in this respect. In our case this caveat applies not only to the *evaluates-to* predicate above, but also to the reflexive transitive closure of immediate dominance, $\blacktriangleleft^\star$.

For the latter case, a standard solution exists, at least in *weak* MSO, the monadic second-order logic on *finite* structures. It is a well-known fact (cf., e.g., Courcelle 1990) that the reflexive transitive closure R^* of a binary relation R on nodes is (weakly) MSO-definable, if R itself is. This is done via a second-order property which holds of the sets of nodes which are closed under R:

(14) $R\text{-}\textit{closed}(X) \equiv \forall x, y\ (x \in X \land R(x, y)\ \rightarrow\ y \in X)$

Now, for any node n, the intersection of all such sets which contain n is exactly the set of m, such that $R^*(n, m)$. Since we are dealing with the (necessarily finite) trees generated by a context-free grammar, this construction can be safely exploited for our purposes.

Then, assuming that $\blacktriangleleft$ is MSO definable, $\blacktriangleleft^\star$ and $\blacktriangleleft^+$ can be defined as follows:

(15)
Dominance:

$$x \blacktriangleleft^\star y \equiv \forall X\,(\blacktriangleleft\text{-}closed(X) \land x \in X \;\rightarrow\; y \in X)$$

Proper dominance:

$$x \blacktriangleleft^+ y \equiv x \blacktriangleleft^\star y \land x \not\approx y$$

To ensure the MSO definability of $\blacktriangleleft$, we make use of a recent variation (Bloem and Engelfriet 1997) of the concept of a tree-walking automaton introduced by Aho and Ullman (1971).

(16)

Definition. A *tree-walking automaton (with* MSO *tests)* is a finite automaton $\mathfrak{A} = (Q, \Delta, \delta, I, F)$ with states Q, directives Δ, transitions δ and the initial and final states $I \subseteq Q$ and $F \subseteq Q$ which traverses a tree along connected edges using three kinds of directives: $\uparrow_i$—“move up to the mother of the current node (if it has one and it is its i-th daughter)”, $\downarrow_i$—“move to the i-th daughter of the current node (if it exists)”, and $\varphi(x)$—“verify that φ holds at the current node”.

For any tree t, such an automaton $\mathfrak{A}$ computes a node relation

(17) $R_t(\mathfrak{A}) = \{(x, y) \mid$ *there is a successful run of* $\mathfrak{A}$ *on* t *starting in an initial state at node* x *and ending at node* y *in a final one*$\}$.

If all the tests $\varphi(x)$ of $\mathfrak{A}$ are MSO definable, $\mathfrak{A}$ specifies a *regular tree-node relation*. Bloem and Engelfriet (1997), who should be consulted for details, prove that any regular tree-node relation is itself MSO definable and provide a general translation of $\mathfrak{A}$ into an MSO formula.

It should be evident, that $\blacktriangleleft$, and, despite its recursive nature, in particular the relation *evaluates-to*, is a regular tree-node relation. In fact, to define it, we can confine ourselves to a *basic* tree-walking automaton which restricts φ to simple tests of node labels. To keep the automaton small and perspicuous, we consider the probably only linguistically relevant case with ternary (binary) branching in the explicit (intended) trees and macros with at most two parameters. The extension to the general case, however, is trivial, as long as branching is bounded by some $n < \omega$.

We define $\mathfrak{A}_\blacktriangleleft = (Q, \Delta, \delta, I, F)$ as follows:

(18)

$$Q = \{q_{in}, e, d, d_1, d_2, d_3, u, u_1, u_2, u_3, q_{fin}\} \qquad I = \{q_{in}\}$$

$$\Delta = \{\downarrow_1, \downarrow_2, \downarrow_3, \uparrow_1, \uparrow_2, \uparrow_3, \mathfrak{L}(x), \mathfrak{C}(x), \Pi_1(x), \Pi_2(x)\} \qquad F = \{q_{fin}\}$$

$$\begin{array}{ll}
\delta = \{ & (q_{in} \multimap\langle\, \mathfrak{L}(x) \,\rangle\!\!\rightarrowtail u), \\
 & (u \multimap\langle \uparrow_1 \rangle\!\!\rightarrowtail d), (d \multimap\langle \downarrow_2 \rangle\!\!\rightarrowtail e), (d \multimap\langle \downarrow_3 \rangle\!\!\rightarrowtail e), \\
\text{case } \mathfrak{L}: & (e \multimap\langle\, \mathbf{L}(x) \,\rangle\!\!\rightarrowtail q_{fin}), \\
\text{case } \mathfrak{C}: & (e \multimap\langle\, \mathbf{C}(x) \,\rangle\!\!\rightarrowtail d_1), (d_1 \multimap\langle \downarrow_1 \rangle\!\!\rightarrowtail e), \\
\text{case } \Pi_1: & (e \multimap\langle\, \Pi_1(x) \,\rangle\!\!\rightarrowtail u_1), (u_1 \multimap\langle \uparrow_2 \rangle\!\!\rightarrowtail u_1), (u_1 \multimap\langle \uparrow_3 \rangle\!\!\rightarrowtail u_1), \\
 & (u_1 \multimap\langle \uparrow_1 \rangle\!\!\rightarrowtail d_2), (d_2 \multimap\langle \downarrow_2 \rangle\!\!\rightarrowtail e), \\
\text{case } \Pi_2: & (e \multimap\langle\, \Pi_2(x) \,\rangle\!\!\rightarrowtail u_2), (u_2 \multimap\langle \uparrow_2 \rangle\!\!\rightarrowtail u_2), (u_2 \multimap\langle \uparrow_3 \rangle\!\!\rightarrowtail u_2), \\
 & (u_2 \multimap\langle \uparrow_1 \rangle\!\!\rightarrowtail d_3), (d_3 \multimap\langle \downarrow_3 \rangle\!\!\rightarrowtail e) \quad \}
\end{array}$$

Since they are constructed pretty much in parallel, it is not hard to convince oneself that on any tree T^L the node relation $R_{T^L}(\mathfrak{A}_{\blacktriangleleft})$ (18) computes under interpretation (17) is identical to the relation (13), appropriately restricted in branching and number of parameters, defines on it.

However, there is yet another interpretation of such an automaton. Viewed as an ordinary finite-state automaton over the alphabet Δ, $\mathfrak{A}$ recognizes a regular (string-)language, the *walking language* W. In our case this is the language $W_{\blacktriangleleft} = \mathfrak{L}(x) \cdot \uparrow_1 \cdot (\downarrow_2 \cup \downarrow_3) \cdot (W_{\mathfrak{C}} \cup W_{\Pi_1} \cup W_{\Pi_2})^* \cdot \mathfrak{L}(x)$[10] with

$$\begin{aligned}
W_{\mathfrak{C}} &= \mathfrak{C}(x) \cdot \downarrow_1 \\
W_{\Pi_1} &= \Pi_1(x) \;\cdot\; (\uparrow_2 \cup \uparrow_3)^* \cdot \uparrow_1 \cdot \downarrow_2 \qquad (19)\\
W_{\Pi_1} &= \Pi_2(x) \;\cdot\; (\uparrow_2 \cup \uparrow_3)^* \cdot \uparrow_1 \cdot \downarrow_3
\end{aligned}$$

It is this walking language which is finally translated into a MSO-formula $trans_{W_{\blacktriangleleft}}(x, y)$. The rather tedious process, which we won't carry through here, proceeds inductively via the following steps:

$$\begin{aligned}
trans_{\emptyset}(x, y) &= \bot \\
trans_{\downarrow_i}(x, y) &= edge_i(x, y) \\
trans_{\uparrow_i}(x, y) &= edge_i(y, x) \\
trans_{\varphi(x)}(x, y) &= \varphi(x) \wedge x = y \\
trans_{W_1 \cup W_2}(x, y) &= trans_{W_1}(x, y) \vee trans_{W_2}(x, y) \qquad (20)\\
trans_{W_1 \cdot W_2}(x, y) &= \exists z\, (trans_{W_1}(x, z) \wedge trans_{W_2}(z, y)) \\
trans_{W^*}(x, y) &= trans^*_W(x, y) \\
trans^*_W(x, y) &= \forall X\, (\forall v, w\, (v \in X \wedge trans_W(v, w) \;\rightarrow\; w \in X) \\
&\qquad \wedge x \in X \;\rightarrow\; y \in X)
\end{aligned}$$

[10]Since Π-nodes are never leftmost daughters, the sublanguage $\mathfrak{L}(x) \cdot \uparrow_1 \cdot (\downarrow_2 \cup \downarrow_3) \cdot (W_{\Pi_1} \cup W_{\Pi_2})^* \cdot W_{\mathbf{C}}{}^* \cdot \mathfrak{L}(x)$ would be even more accurate for our purposes. The trivial changes to $A_{\blacktriangleleft}$ required to compute exactly this language (adding an extra state and duplicating several transitions) just enlarge the automaton, decreasing its perspicuity without providing any new insights.

The resulting formula uses only the $edge_n$ relations, the MSO-definable tests of the original automaton, and—for the Kleene-*-case—the *closed* sets constructed via (14). No recursion is involved: $\blacktriangleleft$ is MSO-definable:

$$x \blacktriangleleft y \equiv trans_{W_{\blacktriangleleft}}(x, y) \tag{21}$$

This approach can be extended to the other members of the *dominance* family of relations: $\mathfrak{A}_{\blacktriangleleft^+}$ is just $\mathfrak{A}_{\blacktriangleleft}$ with the transition $(e -\langle\, \mathbf{L}(x)\,\rangle\!\rightarrow\ u)$ added to δ, and $\mathfrak{A}_{\blacktriangleleft^\star}$ is $\mathfrak{A}_{\blacktriangleleft^+}$ with $F = \{u\}$, and q_{fin}, as well as any transition referring to it, removed from Q and δ, respectively. The inductive translations of the corresponding walking languages $trans_{W_{\blacktriangleleft^\star}}$ and $trans_{W_{\blacktriangleleft^+}}$ into MSO provide an (equivalent) alternative to the definitions in (15). Moreover, while standardly "linguistic" relations like *c-command* or *government* would be defined in terms of *dominance*, this method allows the alternative route of taking, in the spirit of Frank and Vijay-Shanker (1998), *c-command* as the primitive relation of linguistic structure by defining, in a similar, though—since Chomsky's (1985) distinction between *segments* and *categories* has to be accommodated (cf. section 3.2)—somewhat more complicated fashion, an automaton $\mathfrak{A}_{c\text{-}command}$, which computes the intended *c-command* relation directly, without recourse to dominance. We will not, however, pursue this matter here.

3.2. **Relations on categories.** The members of the $\blacktriangleleft$-family of relations are again *geometric* relations: they express dependencies between the *nodes* of the intended trees. Linguistic theory, however, is concerned with the relations between *categories* in the sense of Chomsky (1985:7), i.e., abstract linguistic entities corresponding to *sets of nodes* (their "segments"). These categories induce two families of $\blacktriangleleft^\star$-like relations, $\lhd^\star_C$ ("containment") which holds of two categories α, β if *some* segment of $\alpha \blacktriangleleft^\star$-s (all segments of) β and $\lhd^\star_I$ ("inclusion", generally referred to as dominance) requiring that *all* segments of $\alpha \blacktriangleleft^\star \beta$. One way of obtaining these relations is to identify the categories with their topmost or bottom elements (w.r.t. $\blacktriangleleft^\star$). The restriction of $\blacktriangleleft^\star$ to just these nodes yields $\lhd^\star_C$ and $\lhd^\star_I$, respectively (cf. Kolb 1995, for details). So, as far as linguistic meaningfulness is concerned $\blacktriangleleft$ $(=\lhd)$ is just an auxiliary notion on the intended trees, much in the way $\lhd_L$ is on the LIFTed trees. Since we have $\lhd_L$, $\blacktriangleleft$ should be eliminable. However, this turns out to be not true in any interesting sense: identifying the segments of a category—and especially its top or bottom element—will always involve what amounts to the calculation of $\blacktriangleleft$. What is possible, however,—thanks to our second order language—is to actually make sense of the intuition that linguistic relations "are about" categories and not nodes, by defining, much in the spirit of Rogers, a second order predicate *category*(X) which holds of a set X of nodes in the tree iff X contains all and only the segments of one category, and, contrary to Rogers, the second order predicates $X \triangleleft_C Y$ ($X \triangleleft_I Y$) which hold of sets of nodes X and Y iff X and Y are categories and the maximal (minimal) segment of X $\triangleleft$-s all $y \in Y$. It is easily verified that defined this way $\triangleleft_C$ and $\triangleleft_I$ are tree-orders on the sets of maximal and minimal segments, respectively, and as such are not equivalent to Rogers' definitions of *Includes* and *Excludes*. *C-command* and *m-command*, the other central structural notions of generative grammar can be adapted accordingly.

So we *can* talk about *linguistic* relations—as long as they don't involve precedence—on the derived trees directly. But in confining ourselves to these familiar descriptive means, we put on the straight-jacket of context-freeness again. To make use of the additional power the LIFTed structures provide, we have to find a way to relate the extra-linguistic information they contain to linguistic theory. This issue will be the topic of the following section.

4. LINGUISTIC DERIVATIONS AND EXPLICIT ENCODING

As hinted at in footnote 7, the extralinguistic information contained in the structures generated by LIFTed context-free (macro-)tree grammars, and especially the distribution of the "projection" symbols, does provide an additional layer of control which is responsible for the additional expressivity (under the homomorphism h) exhibited by these grammars. Actually, these elements encode in a rather straightforward way the derivation history of the structures—a fact that allowed us to establish the crucial truth (11e) in the previous section. So, in fact, Mönnich (1999)'s construction extends the expressivity of MSO in yet another way: it allows for a non-trivial treatment of derivations—at least of a certain form.

An application of a tree grammar rule can be viewed as an instance of a *generalized transformation* in the sense of Chomsky (1975). And while the rules of a context-free tree grammar correspond to a rather powerful type of generalized transformations, the derivation steps of the LIFTed, regular tree grammar viewed as transformations, simply replacing one single node in a tree by some other tree, are of the same sort as the basic structure-building operation of the *minimalist* flavor of generative grammar, the restricted generalized transformation scheme MERGE. Like the derivation of an intended tree from a LIFTed tree grammar, a minimalist derivation is a two-story process of MERGing subtrees and rearranging the result by the operation ATTRACT/MOVE in a way completely determined by the distribution of some auxiliary symbols.

There are important differences, however. A standard derivation of a LIFTed tree grammar, proceeds top-down and—more importantly—the homomorphism mapping the explicit structures onto the intended ones operates on the fully specified, terminal structures generated by the grammar. And, as stressed before, it is the existence of these explicit structures which ensures MSO-definability at all. A minimalist derivation, on the other hand, proceeds in a strictly bottom-up fashion, interleaving with the applications of MERGE certain instances of ATTRACT/MOVE, partially evaluating the ATTRACT/MOVE-mapping in the process of assembling the structure, as it were.[11]

Still, restricting ourselves for the moment to overt, i.e. pre-SPELLOUT, head-movement as exemplified by our simplistic treatment of verb-raising in section 2, the similarities between a minimalist and an approach using a LIFTed tree-grammar seem to prevail. Consider again the macro-tree grammar (6) and its sample derivation (figure 3): The derivations from this grammar, though proceeding strictly top-down on their derivation tree, actually build up the terminal trees in a bottom-up fashion, starting from the most deeply embedded VP and working their way up to the final IP, much like the corresponding minimalist derivation. In particular, the macro $+V$ triggers raising of the head just like the $+\nu$ feature does in figure 2. So, as a first approximation, the names of non-trivial macros, i.e. macros with parameters, correspond to strong features, while the application of a macro-rule corresponds to a combined MERGE and ATTRACT/MOVE operation. What lacks any obvious correspondence, however, is the tight top-down control the macro grammar exerts, ensuring that the final structure is "implied" from the beginning, as exemplified by the *VR*-rule in (6). This rule amounts to the introduction of a strong feature independent from the lexical or functional element carrying it, or—more accurately but even more exotic—to a "syntactification" of Brody (1995)'s notion of *pre-syntactic chain*.

Neither a terminal tree generated by some macro-tree grammar nor its minimalist counterpart contains a record of its derivation history.[12] If one takes the LIFTed variant into account, however, the situation changes: To be sure, the triggering macro-symbols are gone here, too, but just to be replaced by an explicit record of their effect. Loosely speaking, the $c \in \mathfrak{C}$ mark instances of MERGE while the $\pi \in \Pi$ seem to encode movements. Even the top-down evaluation of

[11] An additional complication is the difference in in behavior of ATTRACT/MOVE *before* and *after* a certain point in the derivation, SPELLOUT, which corresponds (roughly) to a terminal tree. Post-SPELLOUT ATTRACT/MOVE induces a global mapping akin to the homomorphic mapping producing the intended structures from the explicit ones. Furthermore, the domains of pre- and post-SPELLOUT ATTRACT/MOVE are different: while pre-SPELLOUT ATTRACT/MOVE operates on full subtrees, its post-SPELLOUT counterpart can be viewed as an operation on labels ("features") only.

[12] At least no straightforward one. Even though a minimalist tree necessarily contains traces, they are of little help. Despite Chomsky's insistence on CHAINS as "first-class citizens" of his theory, there is no decent way to capture the notion in his 1995 framework.

the homomorphism h onto the intended structures corresponds to a bottom-up operation in a direct derivation.[13] But, unfortunately, this is where the analogy ends. In (8), the actual "targeting" and "raising" is achieved by the "syntactified presyntactic chains" of the macro-rules and not by h at all. The "movement" involved in the evaluation of h, i.e., the replacement of the projection symbols π by the value of the respective projection functions corresponds, at best, to half of an ATTRACT/MOVE operation, and to the most straight-forward half at that—putting a pre-selected element into the right slot. And this turns out to be not an artifact of the specific grammar (6), either: While the parameterization of operatives provides additional flexibility in *assembling* structures there is no way to *change* a structure already constructed other than by adding to it at operative nodes. In particular, there is no extraction from such a tree. So at least for the standard conception of a derivationalist syntactic theory, our attempt to interpret not only structural relations but derivations directly on explicit trees is doomed to fail.[14]

Yet, the problems we have encountered also indicate the direction towards a possible solution: What we have to look for is yet another explicit encoding of the relevant structures exhibiting a richer internal structure than just composition symbols and placeholders. Evidently, the mapping onto the intended structures will be more involved than the homomorphism we were assuming so far, but as long as an effective mapping can be given at all, the essence of Mönnich (1999)'s argument carries over: It is possible to talk about non-recognizable structures using MSO.

The obvious candidates are the *T-markers* of the minimalist derivations, i.e., trees with leaves labeled by lexical and functional items and all non-terminal nodes carrying rule-names as labels. Our verb-raising example (1) yields the T-maker in figure 6 (ignoring, as we have done throughout, any other movements possibly involved in the minimalist derivation of this sentence).

The problem of the relation between a T-marker and its corresponding derivation is trivial: the T-marker is just a notational means to represent a derivation. Moreover, the set of *T-markers* over a given signature (i.e., the lexicon plus the

[13] cf. the (geometric) hierarchical order of the explicit structures which is almost reversed compared to the intended trees.

[14] As should be clear from the preceding discussion matters look not as bleak if one considers a Brody (1995)-type theory: With presyntactic chains a small number of axioms ensuring the appropriate structural relations between chain elements should suffice to characterize the basic configurations with the pidgin-hole mechanism of the projection functions adding the power required for non-context-free languages. However, as Brody never fails to stress, this leads, in essence, to a representational, static theory.

(. . . dass) Maria Karl
schwimmen lehrt
*‘(. . . that) Mary teaches
Charles to swim’*

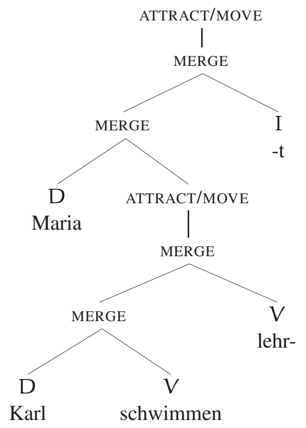


FIGURE 6. A T-marker

set of functional categories) is recognizable.[15] And there is an effective—even though rather complex—mapping into the intended trees obtained by traversing the explicit T-markers from the leaves upward and carrying out the operations denoted by the (non-terminal) node labels, mimicking the actual steps of the minimalist derivation process. This mapping can be characterized in form of a bottom-up tree-transducer.

So all ingredients for a successful pursuit of the original approach are there. However, whether a direct $L^2_{K,P}$ definition on T-markers of the structural relations holding on the intended trees is feasible depends largely on the exact nature of the tree-transducer required, which is subject to further research. Given the type of feature geometry involved in minimalist derivations, however, it seems safe to conjecture that at least for dominance such an undertaking is feasible.

5. CONCLUSIONS

We have shown that the basic structural relations between the constituents of the syntactic structures modern generative grammar assumes can be reconstructed on the explicit trees characterized by a regular tree grammar LIFTed

[15]. . . even though actually constructing a context-free grammar generating it is a tedious process yielding a rather large result. Several of the worst-case calculations Johnson and Lappin (1996) have performed for stepwise minimalist derivations carry over to this construction.

from some context-free (macro-)grammar which generates the intended structures. Yet to make use of Mönnich's (1999) construction in the linguistic metatheory directly, more is needed, *viz.* a notion of how the linguistic "grammar"—given as a set of minimalist operations over a lexicon—relates to these explicit structures.

Yet even though every set of structures posited by linguistic theory and generated by minimalist means can be characterized by some context-free (macro-)tree grammar, the explicit structures obtained from the LIFTed grammar turned out to be not rich enough to relate them in a straightforward manner to the corresponding linguistic derivations. In short, Mönnich's result ensures that we can talk about the *results* of minimalist derivations, but not about the way they are obtained. But since the gist of the minimalist approach to syntactic theory lies in the derivations it admits or excludes, this conclusion disqualifies the direct use of macro-grammar derived structures for metatheoretical purposes. However, the general construction remains valid, and carried out on a different type of structures, in particular on the minimalist T-markers directly, can still be expected to yield the required results.

References

Aho, A. V. and Ullman, J. D. (1971). Translations on a context-free grammar, *Information and Control* **19**: 439–475.

Bloem, R. and Engelfriet, J. (1997). Characterization of properties and relations defined in Monadic Second Order logic on the nodes of trees, *Technical Report 97-03*, Dept. of Computer Science, Leiden University.

Brody, M. (1995). *Lexico-Logical Form*, MIT Press, Cambridge, Ma.

Chomsky, N. (1975). *The Logical Structure of Linguistic Theory*, Plenum, New York. (also 1985, UChicago Press).

Chomsky, N. (1985). *Barriers*, MIT Press, Cambridge, MA.

Chomsky, N. (1995). *The Minimalist Program*, MIT Press, Cambridge, MA.

Courcelle, B. (1990). Graph rewriting: An algebraic and logic approach, *in* J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Vol. B, Elsevier, pp. 193–242.

Doner, J. E. (1970). Tree acceptors and some of their applications, *J. Comput. System Sci.* **4**: 406–451.

Ebbinghaus, H.-D., Flum, J. and Thomas, W. (1996). *Einführung in die mathematische Logik*, 4 edn, Spektrum, Heidelberg.

Fischer, M. J. (1968). Grammars with macro-like productions, *Proceedings of the 9th Annual Symposium on Switching and Automata Theory*, IEEE, pp. 131–142.

Frank, R. and Vijay-Shanker, K. (1998). Primitive c-command, Ms., Johns Hopkins University & Universtiy of Delaware.

Huybregts, M. A. C. (1976). Overlapping dependencies in dutch, *Utrecht Working Papers in Linguistics* **1**: 24–65.

Huybregts, M. A. C. (1984). The weak adequacy of context-free phrase structure grammar, *in* G. J. de Haan, M. Trommelen and W. Zonneveld (eds), *Van periferie naar kern*, Foris, Dordrecht, pp. 81–99.

Johnson, D. and Lappin, S. (1996). A critique of the minimalist program, Ms., IBM Watson Research Center and SOAS.

Kayne, R. (1994). *The Antisymmetry of Syntax*, MIT Press, Cambridge, Ma.

Kolb, H.-P. (1995). Segments, categories, and other dangerous things, Ms., also in: Kolb (1997), pp. 13-26.

Kolb, H.-P. (1997). GB Blues: Two essays on procedures and structures in generative grammar, *Bericht 110*, Arbeitspapiere des SFB 340, Tübingen/Stuttgart.

Maibaum, T. S. E. (1974). A generalized approach to formal languages, *J. Comput. System Sci.* **8**: 409–439.

Mezei, J. and Wright, J. B. (1967). Algebraic automata and contextfree sets, *Inform. and Control* **11**: 3–29.

Miller, P. H. (1991). Scandinavian extraction phenomena revisited: Weak and strong generative capacity, *Linguistics & Philosophy* **14**: 101–113.

Mönnich, U. (1999). On cloning contextfreeness, *This volume*, pp. ???–???

Rabin, M. O. (1969). Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* **141**: 1–35.

Rogers, J. (1994). *Studies in the Logic of Trees with Applications to Grammar Formalisms*, PhD thesis, University of Delaware.

Rounds, W. C. (1970). Tree-oriented proofs of some theorems on context-free and indexed languages, *Proceedings of the 2nd Annual ACM Symposium on Theory of Computing*, pp. 109–116.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language, *Linguistics & Philosophy* **8**(3): 333–344.

Thatcher, J. W. and Wright, J. B. (1968). Generalized finite automata theory with an application to a decision problem of second-order logic, *Math. Systems Theory* **2**(1): 57–81.

SfS, Tübingen University, Wilhelmstr. 113, D-72074 Tübingen
E-mail address: `kolb@sfs.nphil.uni-tuebingen.de`