1

Sidewards without copying

Edward P. Stabler

A traditional movement step relates a single source position to a single c-commanding target position, and never moves an argument to another argument position. But head movement involves non-c-command relations, and control relates two argument positions that are not always in a c-command relation. Special mechanisms could be invoked for these things, but a different strategy slightly generalizes movement and enforces certain fundamental symmetries observed by all movements to block overgeneration. This paper defines a class of 'sideward movement grammars' (SMMGs) with such symmetries, with example applications to adjunct control and head movement. These grammars allow copying, but the question of whether to copy is completely independent of the question of whether to allow sideward movement. Furthermore, since these grammars distinguish complement attachments from others, a simple CED-like constraint can block extractions from specifiers and adjuncts except in the exceptional circumstance of adjunct control. SMMG definable languages are all PMCFG definable, and hence are efficiently recognizable.

1.1 Introduction

One of the most basic properties of human language is its simple, recursive, layered character in which similar structure is iterated, sometimes with special variations at the top, matrix level and at the deepest levels:

Does Alice know that 3	Bob thinks that 2	Carol says 1	you like her? 0

FG-2006.
Organizing Committee:, Paola Monachesi, Gerald Penn, Giorgio Satta, Shuly Wintner.

Certain kinds of recursive symmetry in languages allow the 'pumping lemmas' which have been valuable diagnostics of the availability of certain kinds of grammars. A regular grammar for a language is only possible when the language has a simple symmetry of this kind; context free grammars have a weaker requirement, and so on through the hierarchy of multiple context free languages (Seki et al., 1991), etc.

Many descriptions of human languages involve rearranging constituents. In grammars with movements, how is the structure of each 'layer' affected? This fundamental question is a topic of active study. In early transformational grammars, a set of base structures is generated and then transformed into surface structures, as in the following example (with e and t unpronounced):

[I [know [*e* [I [*e* [saw [who]]]]]]] $\longrightarrow$ [I [know [who [I [*t* [saw [*t*]]]]]].

The sequences of positions related by movement in these accounts are not random. Among other things, landing sites of movement do not disrupt layer structure too much ('structure preservation', 'shape conservation'), and when an element moves through several clauses, it never moves from a high position in a lower clause to a lower position in a higher clause (cf. the 'ban on improper movement' 'chain uniformity', 'level embedding'). So in effect, the hierarchy of each layer of phrase structure is respected in sequences of movements too, another reflection of the basic invariants mentioned at the outset.

Some recent grammars compose generation and transformation steps,[1] so transformations are, in effect, executed as soon as requisite structure is built, reducing the need for revising completed structure:

1.	[saw]+[who]	$\xrightarrow{\text{merge}}$	[saw [who]]
2.	[saw [who]]+[I]	$\xrightarrow{\text{merge}}$	[I [saw [who]]]
3.	[I [saw [who]]]	$\xrightarrow{\text{move}}$	[who [I [saw [~~who~~]]]
4.	[know]+[who [I [saw [~~who~~]]]	$\xrightarrow{\text{merge}}$	[know [who [I [saw [~~who~~]]]]
5.	[know [who [I [saw [~~who~~]]]]+[I]	$\xrightarrow{\text{merge}}$	[I [know [who [I [saw [~~who~~]]]]]

But step 3 shows *who* being copied and deleted, revising the structure built by step 2. One response is to say that the syntax simply copies the earlier structure (perhaps only adding a link, a pointer to the embedded *who*), and then a post-syntax "spellout" process determines which copies to pronounce. This pushes the changes to completed structure out of the syntax, by invoking a "spellout" process that is sensitive to much of the same structure that syntactic operations are sensitive to. When two processes seem to be sensitive to the same structure it

[1] Tree transducer composition, 'deforestation', is a common step for reducing program complexity (Kühnemann, 1999, Reuther, 2003, Maneth, 2004).

is a natural hunch that they are really the *same* process. Adopting this perspective instead, we could then say that the depiction of the derivation 1-5 is slightly misleading: when *who* is introduced in step 1, it satisfies a requirement of the verb but is not actually placed in complement position. Rather, it is held out to be placed at the left edge of the embedded clause. This strategy for (not postponing but) eliminating a kind of structural revision is formalized in MGs (Stabler and Keenan, 2003, Frey and Gärtner, 2002, Michaelis, 2001, Harkema, 2001, Lecomte and Retoré, 1999), but MGs do not ban improper movements.

Now consider the coindexed elements in sentences like these:

He$_i$ tries [e_i to succeed]
He$_i$ laughs [before e_i eating]

These 'obligatory control' (OC) relations have enough in common with movement to suggest a uniform treatment (Hornstein, 2006, 2001, 1999, Polinsky and Potsdam, 2002, Bowers, 1973). If we generalize traditional movement so that a subject can move to another subject position even out of an adjunct as in the latter example, the rest of the phrasal construction can remain completely standard. But such movements between unconnected structures must be restricted to avoid unwanted movements, like these for example:

*John$_i$ likes t_i
*The cook they$_i$ like tried [t_i to make them]
*John$_i$ persuaded Mary [t_i to make them]
*John$_i$'s friends prefer [t_i to behave himself]

One critique of movement analyses of control wonders, if sideways movement is allowed, what rules out sideward movement from complements generally (Landau, 2003, p.477). In the present account, the status and restrictions on sideward movement will be clear: sideward movement from complements is impossible.

Another kind of problem is posed by head movements like this:

[-an]+[ustedes [habl- [español]]] → [[habl-an] [ustedes [~~habl-~~ [español]]]]

If we say x c-commands y in a tree iff a sister of x dominates y, then *habl-* does not c-command its original position. Adapting a proposal from Nunes (2001) and Hornstein (2001), in analogy to phrasal movement, we can compute this result without surgery by keeping the head *habl-* out of its projection so that is available for attachment to the appropriate affix. But the indicated assembly of the head and affix with the rest of the projection is more complicated than any of the other (merge,move) rules, looking suspiciously *ad hoc*. An alternative is to, in effect, allow the head to move before it projects its structure. This yields essentially the same result, but by allowing the head to simply move to another projection, allows the construction of the phrase and the selection of that phrase to be completely standard. But obviously this step needs to bring some analog of the traditional head movement constraint (HMC):

*be -s he have ~~be~~ -en making tortillas

Conventional movements relate source constituents with targets that c-command them. In MGs, the same effect is achieved by keeping the sources separate from the target while they wait for their final licensed positions. In this setting, the needed generalization simply allows new, 'disconnected' elements to be *inserted* into an expression. With this generalization of expressions, we need only one feature-checking operation, *merge*. We define 'sideward movement grammars' (SMMGs) in this way. To avoid overgeneralization, we impose a specifier island constraint (SpIC) and also impose a generalized ban on improper movements. Since all phrases other than the matrix clause are either complements or specifiers, SpIC allows extracted phrases to enter a derivation only through complements, though as explained below this constraint is weaker than usual because a complement can be remnant-moved to a specifier without freezing any of its moving elements.

Formal antecedents include tree adjoining grammar (Joshi and Schabes, 1997) and especially the variants proposed for scrambling (Rambow et al., 2001, Rambow, 1994, Kallmeyer, 1999), certain elaborations of pregroup grammars (Stabler, 2004a, Casadio and Lambek, 2002, Buszkowski, 2001), and the minimalist grammars (MGs) already mentioned. The derivations in these formalisms all extend and simplify complexes of possibly discontinuous constituents. But none of them enforces the ban on improper movements, and none of them defines the same class of languages as SMMGs. SMMG languages are not all MCFG definable, but they are all PMCFG-definable (Seki et al., 1991) and hence are polynomially parsable. We conjecture that all PMCFG languages are SMMG definable too.

1.2 Sideward movement grammars

Let Σ be a finite vocabulary, associated with phonetic and semantic properties. The empty sequence is ϵ. Head movement will be triggered by a morphological property that we indicate with hyphens: a preceding hyphen -s indicates that a lexical head is a suffix; a following hyphen s- indicates a prefix; and the affix s can be empty.

A set of syntactic features $\mathbb{F}$ is partitioned into 2 basic kinds: properties -$\mathbb{F}$ and requirements +$\mathbb{F}$. Properties -$\mathbb{F}$ are either persistent -f or not

-$\overline{\text{f}}$. Requirements +$\mathbb{F}$: some simply require agreement +f, others trigger overt movement +$\underline{\text{f}}$, and others trigger overt movement and also leave a copy +$\underline{\underline{\text{f}}}$. As in MGs, we use the types $\mathbb{T} = \{::, :\}$ to indicate lexical and derived expressions, respectively. The *projections* $\mathbb{P} = \Sigma^* \times \mathbb{T} \times \mathbb{F}^*$. The *expressions* $\mathbb{E} = \mathbb{P} \times \wp(\mathbb{P})$. Consider, e.g., the expression

$$(\text{loves:-v}, \{\text{Mary:-}\overline{\text{focus}}, \text{ who:-}\overline{\text{case}}\text{ -wh}\}).$$

To reduce clutter, we often omit some braces and parentheses,

$$\text{loves:-v, Mary:-}\overline{\text{focus}}\text{, who:-}\overline{\text{case}}\text{ -wh.}$$

With this simpler notation, remember that the head of an expression comes first, and the order of remaining elements (if any) is irrelevant.

A *lexicon* is a finite subset of $\Sigma^* \times \{::\} \times (+\mathbb{F}^* \times -\mathbb{F}^+) \times \{\emptyset\}$ with a designated 'start' category f. A lexical item has *category* f iff its first property is -f or -$\overline{\text{f}}$. f *comp-selects* g iff there as a lexical item with category f whose first requirement is +g or +$\underline{\text{g}}$ or +$\underline{\underline{\text{g}}}$. A *cycle* is a sequence $\text{f}_0 \ldots \text{f}_n$ such that f_0 is the start category, f_{i-1} comp-selects f_i (all $0 < i \leq n$), and no feature appears twice. f *cycle-selects* g iff f precedes g in a cycle. A lexicon is *proper* iff whenever -f precedes -g in any lexical item, some lexical item containing -f has category c and some lexical item containing -g has category d, where d cycle-selects c. With this constraint on lexicons, (Proper), we can remain neutral about whether human languages have a universal, fixed clausal structure. A grammar is given by a proper lexicon, generating the structures in the closure of lexicon with respect to the fixed structure building rules. A completed structure is one containing only one syntactic feature, the start category f. The string language is the set of yields of those completed structures.

There are two structure building relations, ins and merge. The partial binary function *ins* applies to pairs of expressions $((p, S), (q, T))$ only if (i) either (q, T) is lexical or $S = \emptyset$, and (ii) match(p, q) is defined. Its value is given by ins$((p, S), (q, T)) = (p, S \cup \{q\} \cup T)$. Condition (i) is our version of SpIC, mentioned above.

The relation merge $\subset \mathbb{E} \times \mathbb{E}$ applies to (p, S) only if there is a unique $q \in S$ such that match(p, q) is defined. Then it takes as value merge$(p, S \cup \{q\}) = (r, (S - q) \cup T)$ for each match$(p, q) = (r, T)$. The uniqueness condition on application of this function is our version of the shortest move constraint (SMC).

The relation match $\subset \mathbb{P} \times \mathbb{P} \times \mathbb{E}$ is given as follows, where $s, t \in \Sigma^*$ are not marked with an initial or final hyphen to trigger head movement, $\alpha, \beta, \gamma \in \mathbb{F}^*$, $\delta \in \mathbb{F}^+$, and $\cdot \in \mathbb{T}$,

Overt movement:

p	q	match(p,q)		
$\text{s::+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}$	$\text{st:}\alpha,\emptyset$	saturated complement	(i)
$\text{s:+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}$	$\text{ts:}\alpha,\emptyset$	saturated specifier	(ii)
$\text{s}\cdot\text{+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}\delta$	$\text{s:}\alpha,\{\text{t:}\delta\}$	moving,unsaturated projection	(iii)
$\text{s::+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-f}$	$\text{st:}\alpha,\emptyset$	final use of -f	(iv)
$\text{s:+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-f}$	$\text{ts:}\alpha,\emptyset$	final use of -f	(v)
$\text{s}\cdot\text{+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-f}\delta$	$\text{s:}\alpha,\{\text{t:}\delta\}$	moving,unsaturated projection	(vi)
$\text{s}\cdot\text{+}\underline{\text{f}}\alpha$	$\text{t}\cdot\text{-f}\beta$	$\text{s:}\alpha,\{\text{t:-f}\beta\}$	moving with -f	(vii)

covert movement:

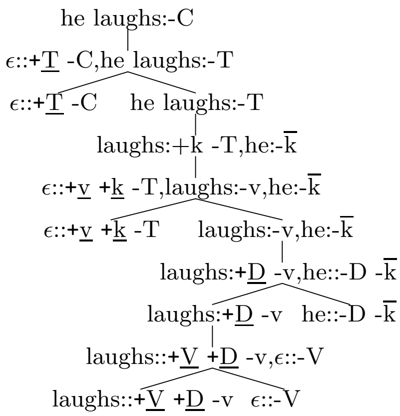
$\text{s}\cdot\text{+f}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}\delta$	$\text{s:}\alpha,\{\text{t:}\delta\}$	check non-persistent $\text{-}\overline{\text{f}}$	(viii)
$\text{s}\cdot\text{+f}\alpha$	$\text{t}\cdot\text{-f}\delta$	$\text{s:}\alpha,\{\text{t:}\delta\}$	final use of -f	(ix)
$\text{s}\cdot\text{+f}\alpha$	$\text{t}\cdot\text{-f}\beta$	$\text{s:}\alpha,\{\text{t:-f}\beta\}$	moving with -f	(x)

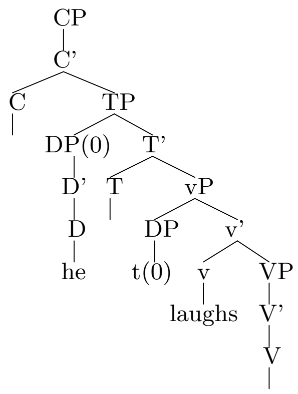
copy movement:

$\text{s::+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}$	$\text{st:}\alpha,\emptyset$	saturated complement	(xi)
$\text{s:+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}$	$\text{ts:}\alpha,\emptyset$	saturated specifier	(xii)
$\text{s::+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}\delta$	$\text{st:}\alpha,\{\text{t:}\delta\}$	moving	(xiii)
$\text{s:+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-}\overline{\text{f}}\delta$	$\text{ts:}\alpha,\{\text{t:}\delta\}$	moving	(xiv)
$\text{s::+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-f}$	$\text{st:}\alpha,\emptyset$	final move to complement	(xv)
$\text{s:+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-f}$	$\text{ts:}\alpha,\emptyset$	final move to specifier	(xvi)
$\text{s::+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-f}$	$\text{st:}\alpha,\{\text{t:-f}\beta\}$	moving with -f	(xvii)
$\text{s:+}\underline{\underline{\text{f}}}\alpha$	$\text{t}\cdot\text{-f}$	$\text{ts:}\alpha,\{\text{t:-f}\beta\}$	moving with -f	(xviii)

We present some examples to illustrate these mechanisms and set the stage for introducing sideward movement.

Example 1: Basics. In the derivation tree on the left, the leaves are lexical items; The binary branches represent applications of insert, and the unary branches, applications of merge.





Note that since insert applies to introduce a projection that can be merged, and the derivation greedily checks features at the earliest possible moment, there is a merge immediately above each insert step. The additional unary branches represent 'external merge' steps: these are the steps that are traditionally called 'movements'. The tree on the right shows the corresponding conventional X-bar structure. It is not difficult to translate the derivations shown here into more traditional depictions like this.[2]

Example 2: Obligatory control into a complement. One idea about obligatory control is that there is a special unpronounced pronoun PRO which, unlike other pronouns, either does not need case or else needs some special kind of case that infinitival tense can assign. But Hornstein argues that the PRO positions can be the empty positions left by movement, as in:

he tries to succeed:-C

ϵ::+$\underline{\text{T}}$ -C,he tries to succeed:-T

ϵ::+$\underline{\text{T}}$ -C   he tries to succeed:-T

tries to succeed:+k -T,he:-$\overline{\text{k}}$

ϵ::+$\underline{\text{v}}$ +$\underline{\text{k}}$ -T,tries to succeed:-v,he:-$\overline{\text{k}}$

ϵ::+$\underline{\text{v}}$ +$\underline{\text{k}}$ -T   tries to succeed:-v,he:-$\overline{\text{k}}$

tries to succeed:+$\underline{\text{D}}$ -v,he:-D -$\overline{\text{k}}$

tries::+$\underline{\text{V}}$ +$\underline{\text{D}}$ -v,to succeed:-V,he:-D -$\overline{\text{k}}$

tries::+$\underline{\text{V}}$ +$\underline{\text{D}}$ -v   to succeed:-V,he:-D -$\overline{\text{k}}$

ϵ::+$\underline{\text{T}}$ -V,to succeed:-T,he:-D -$\overline{\text{k}}$

ϵ::+$\underline{\text{T}}$ -V   to succeed:-T,he:-D -$\overline{\text{k}}$

to::+$\underline{\text{v}}$ -T,succeed:-v,he:-D -$\overline{\text{k}}$

to::+$\underline{\text{v}}$ -T   succeed:-v,he:-D -$\overline{\text{k}}$

succeed:+$\underline{\text{D}}$ -v,he::-D -$\overline{\text{k}}$

succeed:+$\underline{\text{D}}$ -v   he::-D -$\overline{\text{k}}$

ϵ::+$\underline{\text{V}}$ +$\underline{\text{D}}$ -v,succeed::-V

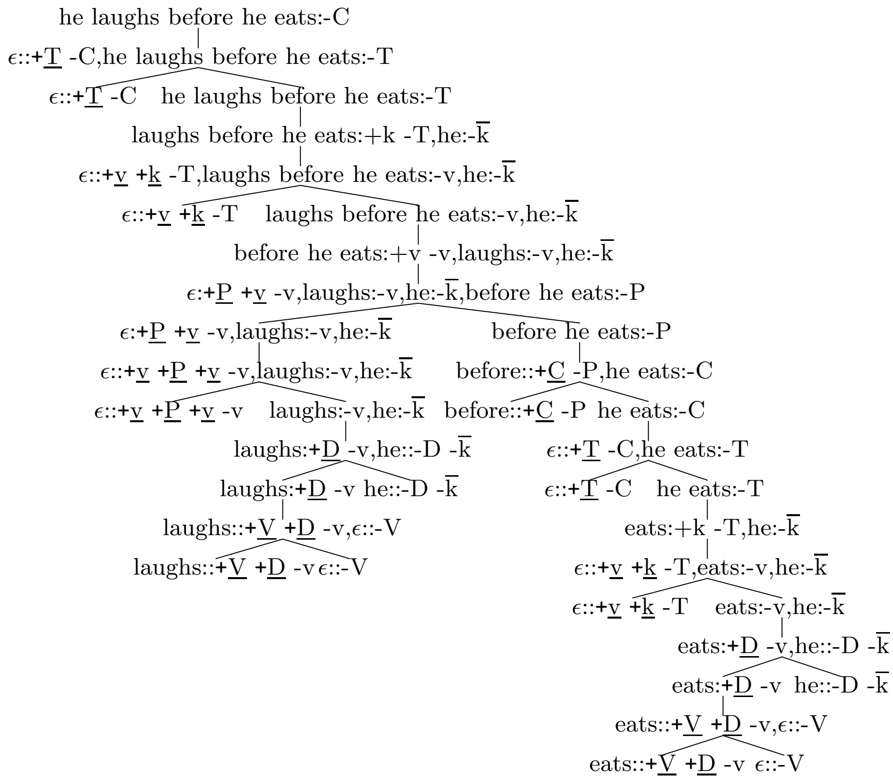
ϵ::+$\underline{\text{V}}$ +$\underline{\text{D}}$ -v   succeed::-V

This derivation is checking the categorial D feature of [he] twice (and then checking its case feature in a higher clausal position, in conformity with Proper). Hornstein suggests that really it is θ-features getting

[2] This translation can be done automatically. See the implementations at `http://www.linguistics.ucla.edu/people/stabler/epssw.htm`.

checked twice in constructions like this. (And there have been suggestions that categorial features generally should be replaced by appropriate complexes of more basic features: θ-features etc.) For present purposes, the simple analysis above provides a suitable starting point.

Example 3: Obligatory control into an adjunct. There are many interesting questions about adjunction, but for present purposes it suffices to adopt a treatment that allows it to be category-preserving, iterable, optional, and opaque to extraction. These properties can be obtained by introducing an empty category to host the adjunct; for clausal adjuncts of noun phrases we use ϵ:+$\underline{\mathrm{N}}$+$\underline{\mathrm{C}}$+$\underline{\mathrm{N}}$-N, and for prepositional modifiers of v we can use: ϵ::+$\underline{\mathrm{v}}$+$\underline{\mathrm{P}}$+$\underline{\mathrm{v}}$-v, as in:



The fact that [before he eats] is a specifier is indicated by the non-lexical status of the selector [ϵ:+$\underline{\mathrm{P}}$ +$\underline{\mathrm{v}}$ -v,laughs:-v,he:-$\overline{\mathrm{k}}$,before he eats:-P]. Since SpIC blocks any extraction from specifiers, we do not need to separately stipulate that adjuncts are islands. So if we introduce right and left X-adjuncts of Y with lexical items of the form ϵ::+$\underline{\mathrm{X}}$+$\underline{\mathrm{Y}}$+$\underline{\mathrm{X}}$-X, or ϵ::+$\underline{\mathrm{X}}$+$\underline{\mathrm{Y}}$-X, respectively (or with any processes that yields similar structure), we get the desired properties for adjuncts: optionality, iterability, and opacity to extraction. This sets the stage for the special treatment of adjunct control.

Since the proposed treatment of adjuncts makes them opaque to extraction, while the proposed treatment of control makes it an extraction relation, we should not get control into adjuncts, but we do:

$$\text{he}_i \text{ laughs before } e_i \text{ eating}$$

Hornstein notices that a slight tweak on our mechanisms can let this kind of case through without allowing other kinds of adjunct extractions. Roughly, if we derive the modifier [before e_i eating,{he}] which wants to attach to a v, and then we derive a v that is looking for a D, we can allow [he] to 'move sideways' onto the v before inserting it into the derivation. This step can be presented in logicians' style, as the inference from the expressions above the line to the one below:

$$\frac{\text{before eating : -P}, \{\text{he : -D-}\overline{\text{k}}\} \qquad \epsilon : \text{+}\underline{\text{v}}\text{+}\underline{\text{P}}\text{+}\underline{\text{v}}\text{-v}, \emptyset \qquad \text{laughs : +}\underline{\text{D}}\text{-v}, \emptyset}{\text{laughs before eating : -v}, \{\text{he : -D-}\overline{\text{k}}\}}$$

We express this step more generally as follows. In a grammar that contains left X-adjuncts of Y, that is, it has some

$$r = \epsilon\text{::+}\underline{\text{X}}\text{+}\underline{\text{Y}}\text{+}\underline{\text{X}}\text{-X}$$

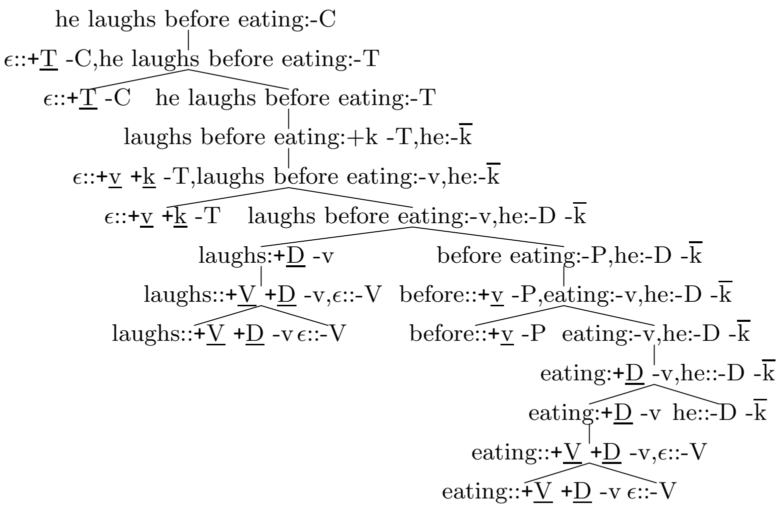
we extend the (ins) relation so that it also applies to $((p, \{a\}), (q, S))$ in the exceptional case where p and q can be chained together by r, using a as follows:

$$\begin{array}{l} \text{match}(q, a) = (b, T), \\ \text{match}(r, b) = (c, U), \\ \text{match}(c, p) = (e, V), \text{ and} \\ \text{match}(e, f) = (g, W) \text{ for } f \in U. \end{array}$$

Notice that the adjoining element r is introduced in the second step to have its 3 initial features checked in sequence. In this special case, let

$$\text{ins}((p, S), (q, T)) = (g, S \cup T \cup (U - \{f\}) \cup V \cup W).$$

Control into right X-adjuncts of Y can be defined similarly, using the lexical item $\ell = \epsilon$::+$\underline{\mathrm{X}}$+$\underline{\mathrm{Y}}$-X, checking its 2 initial features in sequence. With this extension, we obtain:



Example 4: Head movement is similar to adjunct control in relating constituents that do not c-command each other, but, unlike control, we want just the phonetic parts of the heads to move while their projections are developed in their original positions. Nevertheless, there is an application of the sideward movement idea that avoids splitting all phrases kept into triples so that the head can be separate when the phrase is complete, as was done in Stabler (2001).

We extend match so that, when the category of -s::α is comp-selected by t::β and t-s is morphologically well-formed,

p	q	match(p, q)	
-s::α	t::β	ϵ::α,{t-s::β}	suffix left adjoins lower head
s-::α	t::β	ϵ::α,{s-t::β}	prefix right adjoins lower head

And then, when match(q, p) is defined by one of (i-xviii) we bring the adjunction up:

p	q	match(p, q)	
p	q	q,{p}	higher head promoted

With these extensions, we get derivations like the following:

habl- -ϵ -an -ϵ ustedes espanol::-C

habl- -ϵ -an -ϵ::+$\underline{T}$ -C,ustedes espanol::-T

ustedes espanol::-T,habl- -ϵ -an -ϵ::+$\underline{T}$ -C

espanol::+$\underline{k}$ -T,ustedes::-$\overline{k}$,habl- -ϵ -an -ϵ::+$\underline{T}$ -C

ϵ::+$\underline{v}$ +$\underline{k}$ -T,espanol::-v,ustedes::-$\overline{k}$,habl- -ϵ -an -ϵ::+$\underline{T}$ -C

-ϵ::+$\underline{T}$ -C   habl- -ϵ -an::+$\underline{v}$ +$\underline{k}$ -T,espanol::-v,ustedes::-$\overline{k}$

espanol::-v,habl- -ϵ -an::+$\underline{v}$ +$\underline{k}$ -T,ustedes::-$\overline{k}$

espanol::+$\underline{D}$ -v,habl- -ϵ -an::+$\underline{v}$ +$\underline{k}$ -T,ustedes::-D -$\overline{k}$

espanol::+$\underline{D}$ -v,habl- -ϵ -an::+$\underline{v}$ +$\underline{k}$ -T   ustedes::-D -$\overline{k}$

ϵ::+$\underline{V}$ +$\underline{D}$ -v,habl- -ϵ -an::+$\underline{v}$ +$\underline{k}$ -T,espanol:-V

-an::+$\underline{v}$ +$\underline{k}$ -T   habl- -ϵ::+$\underline{V}$ +$\underline{D}$ -v,espanol:-V

espanol:-V,habl- -ϵ::+$\underline{V}$ +$\underline{D}$ -v

ϵ:+$\underline{k}$ -V,espanol::-$\overline{k}$,habl- -ϵ::+$\underline{V}$ +$\underline{D}$ -v

ϵ:+$\underline{D}$ +$\underline{k}$ -V,espanol::-D -$\overline{k}$,habl- -ϵ::+$\underline{V}$ +$\underline{D}$ -v

ϵ::+$\underline{D}$ +$\underline{k}$ -V,habl- -ϵ::+$\underline{V}$ +$\underline{D}$ -v   espanol::-D -$\overline{k}$

-ϵ::+$\underline{V}$ +$\underline{D}$ -v   habl-::+$\underline{D}$ +$\underline{k}$ -V

No revisions of completed structure are needed, and there is no need to treat every phrase as a triple of strings.

1.3 Expressive power and recognition complexity

Previous studies have shown that head movement, though it may seem like a small thing in informal presentations, allows the definition of non-context free patterns even when there is no phrasal movement in the grammar. But the translation from MGs to MCFGs defined by Michaelis (2001) is easily adapted to show that SMMG grammars without copying all define MCFG definable languages. There are various theory-internal arguments for copying in grammar, and various ways to implement them (Stabler, 2004b). See for example Nunes (2001) and Kobele (2006) for some empirical arguments in support of rather powerful copy operations. The addition of copy features makes it easy to define non-semilinear languages like a^{2^n}, but a straightforward extension of Michaelis's translation to these cases shows that they are PMCFG-definable, and hence polynomially recognizable.

1.4 Conclusions

This paper does not attempt to resolve the controversy over whether movement analyses of obligatory control are empirically well-motivated (Landau, 2003, Boeckx and Hornstein, 2004), but provides a formalization of some parts of these ideas that can be rigorously studied.

Although SMMGs can be regarded as extending MGs, notice that they differ in a number of significant respects: (1) SMMGs extend the domain of movement just slightly to offer tightly constrained treatments of obligatory control and head movement. Future work may find ways to make these constraints more general and natural. And there are regularities in the definition of *match* that should allow a more elegant statement. (2) MGs are bound by SMC, while SMMGs also are required to respect SpIC and Proper, and future work may provide further additions. (3) To handle head movement, MGs require either extra rules for head movement (Michaelis, 2001) or else one of the approaches mentioned in the introduction. SMMGs allow head movement with a simple mechanism analogous to the sideward mechanisms used for control. (4) MGs have no copy operation, and while none of the analyses above depend on it, SMMGs allow copying. That is, we have presented a treatment of sideward movement that does not rely in any way on the copy theory of movement for its appeal. In the present setting, sideward movement is a natural option not because we already have operations on copies, but because we already have operations on moving phrases (the original phonetic materials, not copies). SMMGs are naturally extended to allow copying though, setting the stage for studying proposals about overt copying (Boeckx et al., 2005, for example) – unfortunately beyond the scope of this short report. All the mechanisms proposed here are obtained in the well-understood and feasible space of PMCFG-definable languages.

References

Boeckx, Cedric and Norbert Hornstein. 2004. Movement under control. *Linguistic Inquiry* 35(3):431–452.

Boeckx, Cedric, Norbert Hornstein, and Jairo Nunes. 2005. Overt copies in reflexive and control structures: A movement analysis. In *Workshop on New Horizons in the Grammar of Raising and Control, Harvard University*.

Bowers, John S. 1973. *Grammatical Relations*. Ph.D. thesis, Cambridge, Massachusetts, Massachusetts Institute of Technology.

Buszkowski, Wojciech. 2001. Lambek grammars based on pregroups. In P. de Groote, G. Morrill, and C. Retoré, eds., *Logical Aspects of Computational Linguistics*, Lecture Notes in Artificial Intelligence, No. 2099. NY: Springer.

Casadio, Claudia and Joachim Lambek. 2002. A tale of four grammars. *Studia Logica* 71(3):315–329.

Frey, Werner and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings, Formal Grammar'02*. Trento.

Harkema, Henk. 2001. *Parsing Minimalist Languages*. Ph.D. thesis, University of California, Los Angeles.

Hornstein, Norbert. 1999. Movement and control. *Linguistic Inquiry* 30:69–96.

Hornstein, Norbert. 2001. *Move! A Minimalist Theory of Construal*. Oxford: Blackwell.

Hornstein, Norbert. 2006. On control. In R. Hendriks, ed., *Contemporary Grammatical Theory*. Oxford: Blackwell. Forthcoming.

Joshi, Aravind K. and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, eds., *Handbook of Formal Languages, Volume 3: Beyond Words*, pages 69–124. NY: Springer.

Kallmeyer, Laura. 1999. *Tree Description Grammars and Underspecified Representations*. Ph.D. thesis, Universität Tübingen.

Kobele, Gregory M. 2006. Deconstructing copying: Yoruba predicate clefts and universal grammar. Presented at the LSA. http://www.linguistics.ucla.edu/people/grads/kobele/papers.htm.

Kühnemann, Armin. 1999. Comparison of deforestation techniques for functional programs and for tree transducers. In *Fuji International Symposium on Functional and Logic Programming*, pages 114–130.

Landau, Ido. 2003. Movement out of control. *Linguistic Inquiry* 34(3):471–498.

Lecomte, Alain and Christian Retoré. 1999. Towards a minimal logic for minimalist grammars. In *Proceedings, Formal Grammar'99*. Utrecht.

Maneth, Sebastian. 2004. *Models of Tree Translation*. Ph.D. thesis, Universiteit Leiden.

Michaelis, Jens. 2001. *On Formal Properties of Minimalist Grammars*. Ph.D. thesis, Universität Potsdam. *Linguistics in Potsdam 13*, Universitätsbibliothek, Potsdam, Germany.

Nunes, Jairo. 2001. Sideward movement. *Linguistic Inquiry* 32:303–344.

Polinsky, Maria and Eric Potsdam. 2002. Backward control. *Linguistic Inquiry* 33:245–282.

Rambow, Owen. 1994. *Formal and computational aspects of natural language syntax*. Ph.D. thesis, University of Pennsylvania. Computer and Information Science Technical report MS-CIS-94-52 (LINC LAB 278).

Rambow, Owen, K. Vijay-Shanker, and David Weir. 2001. D-tree substitution grammars. *Computational Linguistics* 27(1):87–121.

Reuther, Stefan. 2003. Implementing tree transducer composition for the Glasgow Haskell compiler. Diplomarbeit, Technische Universität Dresden.

Seki, Hiroyuki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science* 88:191–229.

Stabler, Edward P. 2001. Recognizing head movement. In P. de Groote, G. Morrill, and C. Retoré, eds., *Logical Aspects of Computational Linguistics*, Lecture Notes in Artificial Intelligence, No. 2099, pages 254–260. NY: Springer.

Stabler, Edward P. 2004a. Tupled pregroup grammars. UCLA. Available at http://www.linguistics.ucla.edu/people/stabler/epspub.htm.

Stabler, Edward P. 2004b. Varieties of crossing dependencies: Structure dependence and mild context sensitivity. *Cognitive Science* 93(5):699–720.

Stabler, Edward P. and Edward L. Keenan. 2003. Structural similarity. *Theoretical Computer Science* 293:345–363.