Strict Lexicalization of a Grammar

Cem Bozşahin; November 9, 2011
Cognitive Science Department, Middle East Technical University
bozsahin@metu.edu.tr

The difference between a lexicalized phrase structure grammar and a lexicalized combinatory grammar can best be observed from an example. It will highlight the difference between a structure-dependent and type-dependent grammar as well.

Consider the first phrase structure grammar in *Syntactic Structures* below. I kept the original sequence operator '+' for comparison on the same grounds.

(1)	S	→	NP+VP	T	→	the	Chomsky (1957:26)
	NP	→	T+N	N	→	man, ball, etc.	
	VP	→	Verb+NP	Verb	→	hit, took, etc.	

The first rule can be interpreted from the perspective of either constituent of S. From NP's perspective, it is a function that takes a VP to the right to yield S, in present terms $NP{=}S/VP$. From VP's perspective, it is a function that takes an NP to the left and yield S, presently $VP{=}S\backslash NP$. We get the equivalences in the first row below if we apply the same logic to NP and VP rules. The second and third rows show the categories derived by further substitution (which is possible because this is essentially the algebra of concatenation as Lambek 1961 showed. Morrill 2000 points out the same property of Lambek calculus).

(2)	$NP{=}S/VP$ $VP{=}S\backslash NP$	$T{=}NP/N$ $N{=}NP\backslash T$	Verb$=VP/NP$	$NP{=}VP\backslash$ Verb
	$T{=}(S/VP)/N$	$N{=}NP\backslash((S/VP)/N)$	Verb$=(S\backslash NP)/NP$	$NP{=}(S\backslash NP)\backslash((S\backslash NP)/NP)$
	$NP{=}S/(S\backslash NP)$	$N{=}NP\backslash(NP/N)$	$T{=}(S/(S\backslash NP))/N$	$N{=}NP\backslash((S/(S\backslash NP))/N)$

We can find all substitutions for S, NP and VP of (1), and eliminate the entire first column in the grammar. What we cannot eliminate is the second column because that would change the empirical coverage of the grammar. The resulting grammar looks like (3), where the arrow notation is replaced by ':=' because these are no longer rewrite rules.

(3) the := T = $(S/(S\backslash NP))/N$
man, ball, etc. := N = $NP\backslash((S/(S\backslash NP))/N)$
hit, took, etc. := Verb = $(S\backslash NP)/NP$

This is *radical lexicalization*. The term is due to Karttunen (1989), where he attempted to eliminate all phrase structure rules. This style of lexicalization is explicitly endorsed earlier by Lambek (1961) as well. As context-free substitutions, they cannot break the phrase structure grammar barrier as Chomsky observed back in 1957. Breaking the barrier is a necessity because we know that human languages are provably not context-free (Shieber, 1985). There are no Dutch or Swiss-German grammars that can be radically lexicalized this way because of crossing dependencies.

Strict lexicalization is an umbrella term to refer to the full lexicalization of structure-dependent and type-dependent grammars, at least to the level of linear-indexed grammars. Dutch and Swiss-German force us to go up to this level. The *type* is what we've been writing with slashes. It arises from the order semantics of combinators of Schönfinkel (1924), Curry and Feys (1958); see Steedman (2000) for a fuller exposition.

We have so far used only the applicative fragment of combination in substitutions. This is equivalent to using (4a–b) only, which Bar-Hillel et al. proved back in 1960. Now we know that this is empirically not adequate. We have no way to lexicalize a category for strings such as 'NP Verb', which is exactly what is required for lexical items such as *whom*. We have NP=S/VP and Verb=VP/NP, but not 'NP Verb'=S/NP. This category is necessary in the lexicon because it is a possible domain of locality; observe $*[\textit{whom}]_{(N\backslash N)/(S/NP)}\ [\textit{hit the man}]_{(S\backslash NP)}$ versus $[\textit{whom}]_{(N\backslash N)/(S/NP)}\ [\textit{the man hit}]_{(S/NP)}$.

(4) a. Forward Application:
$XY\!: f \quad Y\!: a \quad \rightarrow \quad X\!: fa$ (>)

b. Backward Application:
$Y\!: a \quad XY\!: f \quad \rightarrow \quad X\!: fa$ (<)

c. Forward Composition:
$XY : f \quad YZ : g \quad \rightarrow \quad XZ : \lambda x.f(gx)$ ($>\mathbf{B}$)

d. Backward Composition:
$YZ : g \quad XY : f \quad \rightarrow \quad XZ : \lambda x.f(gx)$ ($<\mathbf{B}$)

e. Forward Crossing Composition:
$XY : f \quad YZ : g \quad \rightarrow \quad XZ : \lambda x.f(gx)$ ($>\mathbf{B}_{\times}$)

f. Backward Crossing Composition:
$YZ : g \quad XY : f \quad \rightarrow \quad XZ : \lambda x.f(gx)$ ($<\mathbf{B}_{\times}$)

g. Forward Type Raising:
$A : a \quad \rightarrow \quad T/(T\backslash A) : \lambda f.fa$ (where *A* is an argument type in the lexicon onto *T*) ($>\mathbf{T}$)

h. Backward Type Raising:
$A : a \quad \rightarrow \quad T\backslash(T/A) : \lambda f.fa$ (where *A* is an argument type in the lexicon onto *T*) ($<\mathbf{T}$)

CCG's radical lexicalization makes these categories available for adoption by any lexical item because of (4c-h). Notice that not only do we get unorthodox categories by this process, we also get their semantics immediately because of the combinator basis of CCG. For example, *the man hit* is derived by type raising of *the man* followed by composition with *hit*, with the category S/NP, as shown in (5).

(5)
$$
\begin{array}{cc}
\text{the man} & \text{hit}\\
\overline{NP} & \overline{(S\backslash NP)/NP}\\
: man' & : \lambda x \lambda y.hit'xy\\
\overline{S/(S\backslash NP)}^{>\mathbf{T}} & \\
: \lambda P.P man' & \\
\multicolumn{2}{c}{\overline{S/NP}^{>\mathbf{B}}}\\
\multicolumn{2}{c}{: \lambda x.hit'x\, man'}
\end{array}
$$

This is an empirical requirement; to radically lexicalize *whom*'s and similar items' domain of locality. Similar considerations follow from radically lexicalizing the domains of locality of parasitic gaps, coordination, and from constituents engendered by intonation, such as focus projection, topicalization, backgrounding etc.; see Steedman (2000), Steedman and Baldridge (2009), Özge and Bozsahin (2010).

References

Bar-Hillel, Y., C. Gaifman, and E. Shamir (1960). On categorial and phrase structure grammars. *The Bulletin of the Research Council of Israel 9F*, 1–16.

Chomsky, N. (1957). *Syntactic Structures*. The Hague: Mouton.

Curry, H. B. and R. Feys (1958). *Combinatory Logic*. Amsterdam: North-Holland.

Karttunen, L. (1989). Radical lexicalism. In M. Baltin and A. Kroch (Eds.), *Alternative Conceptions of Phrase Structure*. Chicago: University of Chicago Press.

Lambek, J. (1961). On the calculus of syntactic types. In R. Jakobson (Ed.), *Structure of Language and Its Mathematical Aspects*, pp. 166–178. American Mathematical Society. Proceedings of Symposia in Applied Mathematics, vol. XII.

Morrill, G. (2000). Incremental processing and acceptability. *Computational Linguistics 26*(3), 319–338.

Özge, U. and C. Bozsahin (2010). Intonation in the grammar of Turkish. *Lingua 120*, 132–175.

Schönfinkel, M. I. (1920/1924). On the building blocks of mathematical logic. In J. v. Heijenoort (Ed.), *From Frege to Gödel*. Harvard University Press, 1967. Prepared first for publication by H. Behmann in 1924.

Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy 8*, 333–343.

Steedman, M. (2000). *The Syntactic Process*. Cambridge, MA: MIT Press.

Steedman, M. and J. Baldridge (2009). Combinatory categorial grammar. In K. Börjars (Ed.), *Non-transformational syntax: a guide to current models*. Blackwell. to appear.