Computational perspectives on phonological constituency and recursion

Kristine M. Yu | University of Massachusetts Amherst | krisyu@linguist.umass.edu

A central claim of computational phonology is that finite state grammars are sufficient to describe phonological patterns in natural language (Kaplan and Kay, 1994; Heinz, 2018). Yet, recursivity in phonology has been taken to imply that finite state grammars are in fact insufficient and that context free grammars are necessary (Schreuder et al 2009, Hulst 2010). While finite state grammars are insufficient, it isn't because they can't be recursive, but because they can't recognize prosodic constituency. Recursion is omnipresent in finite state descriptions of phonology: any finite state automaton with a loop is recursive. Recursion is a fundamental property in descriptions of phonological patterns, independent of what assumptions are made about how or even whether strings are organized into particular prosodic categories. This computational perspective: (i) clarifies that what is at issue in many questions about recursivity in phonology is not recursion per se nor (un)boundedness of embedding, but capturing generalizations in the grammar by recognizing phonological constituents, and (ii) shows that computational complexity isn't a reason to reject recursive categories in favor of independent ones.

Consider the two finite state descriptions in (1) of licit CV-strings in a language with only (C)V syllables. Both grammars are finite state since the rules are restricted to the form A $\rightarrow$ aB, and A $\rightarrow \epsilon$ where A and B are non-terminal categories and a is a terminal symbol, and ϵ is the empty string. Only (1b) has recursion and recognizes a commonality between parts of the string, as evidenced in the loops between states 0 and 1 ($0 \rightarrow C\,1$, $1 \rightarrow V\,0$) and self-loop over state 0 ($0 \rightarrow C\,0$). (1b) can thus generate arbitrarily long strings of the language, while (1a) would need to add an additional state for each additional segment. This contrast illustrates that recursion happens when you recognize that the structure of a constituent is similar to one of its parts. Doing so can make grammars more succinct (Yu, 2018). There is, however, no difference in bracketing in derivations from the two grammars, e.g., in a derivation of VCV, both grammars pick out the right-branching constituency: [V[C[V]]]. (1b) is thus insufficient as a phonological description. The only constituents of length > 1 in finite state grammars are suffixes (or prefixes, for a left-branching grammar). There is no way a finite state grammar could recognize the constituency [V][CV], i.e., "syllables", or other prosodic categories.

The comparison (2) illustrates how context free grammars can recognize generalizations conditioned on prosodic domains. Both grammars generate arbitrarily long strings, and both derivations of twènty-six gòod-looking Jàpanese D́Js in (2) derive the desired recursive φ-phrase constituency structure proposed in Gussenhoven (2005); Ito and Mester (2007). However, the finite state grammar can't recognize the generalization that non-suffixal elements such as *Japanese* are constituents (ωs) with common properties and can't recognize the generalization of prosodic adjunction as a process, which the context-free grammar does with the single rule: $\varphi \rightarrow \omega\,\varphi$. In contrast, the finite-state grammar must include an additional adjunction-like rule for each lexical element.

The ramification of context free rather than finite state descriptions of phonological patterns (what is computed) for the parsing complexity of phonological processes and the syntax-prosody interface (how it could be computed) is minimal, as is the ramification of recursive categories vs. independent categories in a structural analysis. Parsing complexity is not a reason to reject recursive categories (Ito and Mester, 2003) in favor of independent categories. So long as recursion depth has a finite bound (which is always the case in produced sentences and words), it is guaranteed that phonological processes can be computed with finite state methods on strings rather than on trees (Yu, To appear). A finite number of states can be used to mimic traversing a tree to recognize maximal and minimal projections. I demonstrate how parsing tonal and accentual structures in Japanese can be implemented over strings for recursive phonological phrases or Major/Minor Phrases (Ito and Mester, 2013).

(1) Two finite state descriptions of CV-phonotactics

(a) Non-recursive grammar

$0 \to C\,1, 0 \to V\,2, 1 \to V\,2$

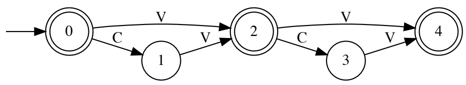
$2 \to C\,3, 2 \to V\,4, 3 \to V\,4, 4 \to \epsilon$

(b) Recursive grammar

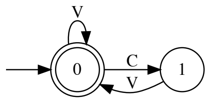
$0 \to V\,0, 0 \to \epsilon$

$0 \to C\,1, 1 \to V\,0$

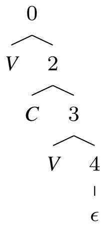
Non-recursive automaton



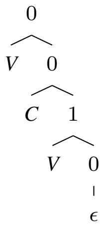
Recursive automaton



Non-recursive derivation of VCV



Recursive derivation of VCV



(2) Two recursive descriptions of phonological structure in the rhythm rule

(a) Finite state grammar

$\varphi \to \textit{twenty-six}\,\varphi, \varphi \to \textit{good-looking}\,\varphi$

$\varphi \to \textit{Japanese}\,\omega$

$\omega \to DJs\,\zeta, \zeta \to \epsilon$

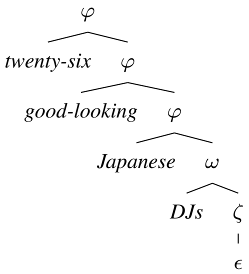
(b) Context-free grammar

$\varphi \to \omega\,\varphi, \varphi \to \omega\,\omega$

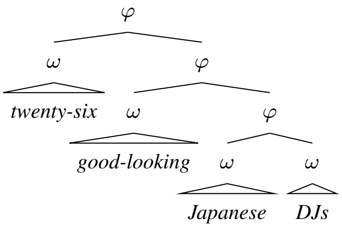
$\omega \to Ft \ldots, Ft \to \sigma \ldots$

$\sigma \to twen, \sigma \to ty, \ldots, \sigma \to Js$

(c) Finite state derivation



(d) Context-free derivation



Gussenhoven, C. 2005. Procliticized phonological phrases in English: Evidence from rhythm. *Studia Linguistica*, 59:174– 193.

Heinz, Jeffrey. 2018. The computational nature of phonological generalizations. In L. M. Hyman and F. Plank, eds., *Phonological typology*. De Gruyter Mouton, Berlin, p. 126–195.

van der Hulst, Harry. 2010. Re recursion. In Harry van der Hulst, editor, *Recursion and human language*. De Gruyter Mouton, Berlin/New York, pages xv–liii.

Ito, Junko and Armin Mester. 2003. Weak layering and word binarity. In T. Honma et al, editors, *A new century of phonology and phonological theory*, pages 26–65.

Ito, J. and A. Mester. 2007. Prosodic categories and recursion. WPSI3, Indiana University.

Ito, Junko and Armin Mester. 2013. Prosodic subcategories in Japanese. *Lingua*, 124:20–30.

Kaplan, R.M. and M. Kay. 1994. Regular models of phonological rule systems. *Comp. Ling.*, 20(3):331–378.

Schreuder, M., et al. 2009. Recursion in phonology. *Lingua*, 119:1243–1252.

Yu, Kristine M. 2018. Advantages of constituency. In *Formal grammar 22 and 23*, Springer-Verlag, Berlin, Heidelberg.

Yu, Kristine M. To appear. Parsing with Minimalist Grammars and prosodic trees. In R. C. Berwick and E. P. Stabler, eds., *Minimalist parsing*. Oxford University Press.