From the lexicon to a stochastic grammar

Michael Becker, University of Massachusetts, Amherst • NELS 38, University of Ottawa

The problem: Getting from a lexicon to a grammar

Phonological processes that are restricted to certain lexical items typically apply stochastically to novel items.

The behavior of novel items reflects lexical trends (Hayes & Londe 2006, Albright & Hayes 2003, Zuraw 2000, and several others)

→ **We need a way to project a stochastic grammar from the lexicon**

Case study: Turkish voicing alternations

Turkish has intervocalic voicing in some words, but not others:

ta**t**	ta**d**-i	‘taste’
kanat	kana**d**-i	‘wing’
vs.		
du**t**	du**t**-u	‘mulberry’
sepe**t**	sepe**t**-i	‘basket’

Factors that correlate with the relative proportion of alternating stops:

Size Monosyllables don’t usually alternate, polysyllables usually do

Place final [t]’s don’t usually alternate, final [p], [k] usually do

Lexical statistics and experimental results that confirm speakers’ knowledge of the pattern are in Becker, Ketrez & Nevins (2007).

How do speakers learn the proportion of alternating stops for each size and place?

The solution: Generalized Cloning

When lexical items demand conflicting rankings, BCD (Prince & Tesar 1999) detects inconsistency and stalls:

	OO-Ident(voice)	*VTV
dut-u ~ dud-u	W	L
tad-ı ~ tat-ı	L	W

Generalized cloning: All clones are lexically specific.

	OO-Ident (voice)$_{\{dut\}}$	*VTV	OO-Ident (voice) $_{\{tat\}}$
dut-u ~ dud-u	W	L	
tad-ı ~ tat-ı		W	L

Result: A categorical grammar for listed lexical items:

OO-Ident(voice)$_{\{dut, ...\}}$ » *VTV » OO-Ident(voice)$_{\{tat, ...\}}$

Generalization:
Lexically-specific grammar → Stochastic grammar

OO-Ident(voice)$_{60\%}$ » *VTV » OO-Ident(voice)$_{40\%}$

Specific constraints first

Speakers keep track of monosyllables independently of polysyllables, thanks to the existence of initial-syllable faithfulness:

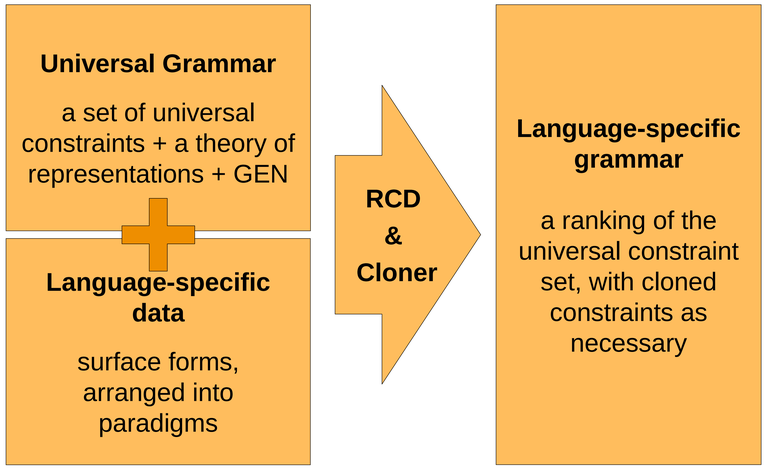
	OO-Ident (voice)$_{\sigma 1}$	OO-Ident (voice)	*VTV
dut-u ~ dud-u	W	W	L
tad-ı ~ tat-ı	L	L	W
sepet-i ~ seped-i		W	L
kanad-ı ~ kanat-ı		L	W

If the learner wrongly clones the general OO-Ident(voice) first, the general constraint will account for all exceptions, and the size effect will not be learned:

☹ OO-Ident(voice)$_{\{dut, sepet, ...\}}$ » *VTV » OO-Ident(voice)$_{\{tat, kanat, ...\}}$

The learner must clone specific constraint first to list monosyllables, then clone the general OO-Ident(voice) to list polysyllables:

OO-Ident(voice)$_{\sigma 1\{dut, ...\}}$,OO-Ident(voice)$_{\{sepet, ...\}}$ » *VTV » OO-Ident(voice)$_{\sigma 1\{tat, ...\}}$, OO-Ident(voice)$_{\{kanat, ...\}}$



The learner

The learner reads in the words of the given language one by one, and runs them through the grammar, creating a candidate set according to principles of OT-CC (McCarthy 2007). If the winner is different from the surface form, a winner-loser pair is formed and submitted to the RCD algorithm.

If RCD detects inconsistency, the learner clones a constraint that assigns the non-zero minimum of both W’s and L’s to the set of inconsistent ERC’s. This continues recursively, until the data becomes consistent, or can’t be made consistent by cloning.

The resulting grammar is categorical relative to existing lexical items, but can apply stochastically to novel items.