Computation, learning, and typology

Class 2: Simplicity

Adam Albright and Eric Baković

CreteLing 2023 — July 2023

creteling2023.phonology.party

Rule format and representations

Recap from last time

- Recipe for a predicted phonological typology
  - Model of grammar
  - Model of transmission/learning
  - (Also, non-linguistic forces)
- Began to sketch how one piece of a grammatical model can generate typological distributions
  - Constraint rankings in OT $\Rightarrow$ factorial typology
- Goal for the next few classes: step back and think more broadly about how properties of formal models shape typological predictions

Possible vs. impossible rules

- SPE-style rewrite rules: A $\rightarrow$ B / C ___ D
- Rewrite notation imposes interesting limits on what can be expressed
- Possible
  - Devoice final obstruents
  - Devoice obstruents before low vowels
- Difficult or impossible
  - Devoice every other obstruent
  - Devoice final obstruents in words that are phonological neighbors of [kæt]

The format of rules

- Rich literature on the computational complexity of languages generated by rules of different formats
- Heinz (2010): hard limits on possible phonological patterns, from formal limits on rules/constraints

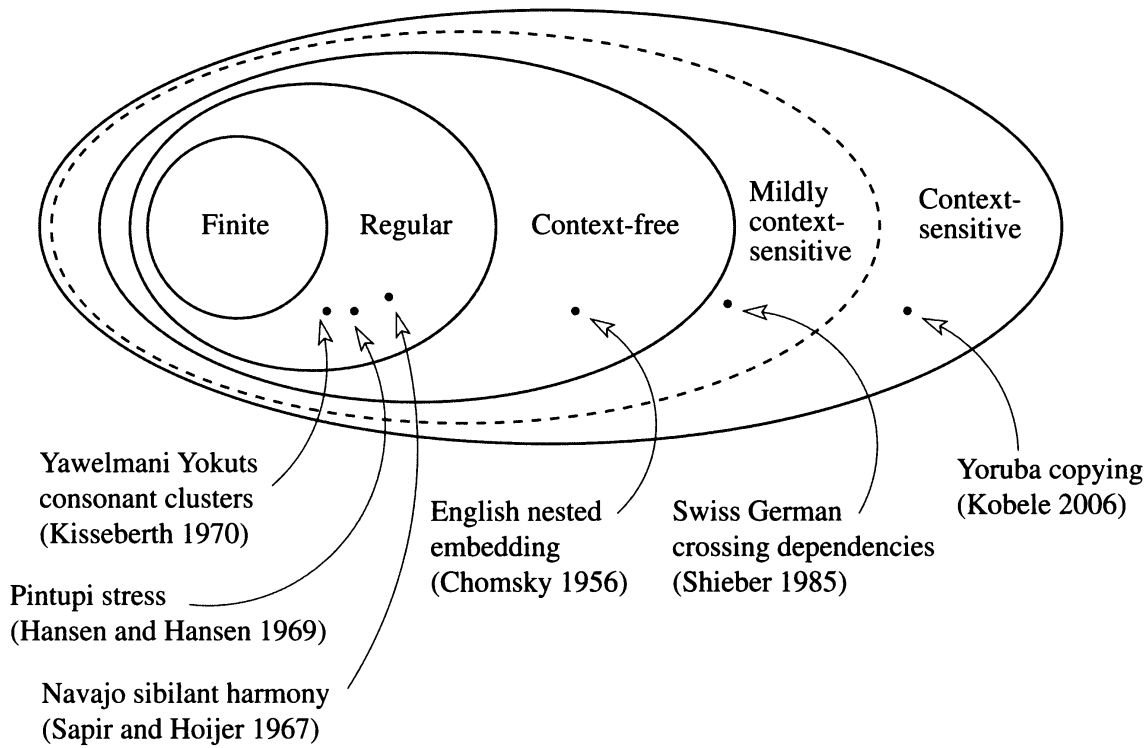

Figure 1
The Chomsky hierarchy

Phonological features

- Phonological features shape how rules/constraints are expressed
- Sizeable literature arguing about feature values and geometries, based on set of targets and contexts of attested processes

Phonological features

- How does our choice of features constrain the types of processes/languages that we can describe?

Phonological features

- How does our choice of features constrain the types of processes/languages that we can describe?
  - Every segment can be described with features
  - Every class can be described as a list of features
  - So, any class can be described

Phonological features

- How does our choice of features constrain the types of processes/languages that we can describe?
  - Every segment can be described with features
  - Every class can be described as a list of features
  - So, any class can be described
- Choice of features affects which classes are simpler to describe
- Literature on features has used typological data to reason about what should be simpler to describe
  - We'll need to scrutinize the link between simplicity and typology more carefully below
  - For the moment: what does it even mean to be simpler?

Simplicity

Simplicity of grammars

- What does it mean for a one grammar to be simpler than another?
- Starting small: what does it mean for one rule/constraint to be simpler than another?

The SPE evaluation metric

Chomsky & Halle (1968), chap. 8: compare two processes, in different languages

Language 1:

- i $\rightarrow$ y / ___ p
- i $\rightarrow$ y / ___ y
- i $\rightarrow$ y / ___ r
- i $\rightarrow$ y / ___ a

Language 2:

- i $\rightarrow$ y / ___ p
- r $\rightarrow$ l / ___ y
- t $\rightarrow$ p / ___ r
- s $\rightarrow$ n / ___ a

- The first can be rewritten: i $\rightarrow$ y / ___ {p,y,r,a}
- Disjunction = 'schema' which expands to the set of rules above

The evaluation metric

Chomsky & Halle (1968):334

The "value" of a sequence of rules is the reciprocal of the number of symbols in the minimal schema that expands to this sequence, where the minimal schema is the one with the smallest number of symbols.

Symbols count features

Chomsky & Halle (1968):335

...it is almost always taken for granted that phonological segments can be grouped into sets that differ as to their “naturalness.” Thus, the sets comprising all vowels or all stops or all continuants are more natural than randomly chosen sets composed of the same number of segment types. No serious discussion of the phonology of a language has ever done without reference to classes such as vowels, stops, or voiceless continuants. On the other hand, any linguist would react with justified skepticism to a grammar that made repeated reference to a class composed of just the four segments [p r y a].

Symbols count features

Chomsky & Halle (1968):335

These judgments of ”naturalness” are supported empirically by the observation that it is the ”natural” classes that are relevant to the formulation of phonological processes in the most varied languages, though there is no logical necessity for this to be the case. In view of this, if a theory of language failed to provide a mechanism for making distinctions between more or less natural classes of segments, this failure would be sufficient reason for rejecting the theory as being incapable of attaining the level of explanatory adequacy

Featural complexity

The disjunctive rule i → y / ___ {p,y,r,a}

(10)

$$\begin{bmatrix} +\text{voc} \\ -\text{cons} \\ +\text{high} \\ -\text{back} \\ -\text{ant} \\ -\text{cor} \\ +\text{cont} \\ -\text{nasal} \\ -\text{strid} \end{bmatrix} \rightarrow \begin{bmatrix} -\text{voc} \\ -\text{cons} \\ +\text{high} \\ -\text{back} \\ -\text{ant} \\ -\text{cor} \\ +\text{cont} \\ -\text{nasal} \\ -\text{strid} \end{bmatrix} \Big/ \_\_ \left\{ \begin{bmatrix} -\text{voc} \\ +\text{cons} \\ -\text{high} \\ -\text{back} \\ +\text{ant} \\ -\text{cor} \\ -\text{cont} \\ -\text{nasal} \\ -\text{strid} \end{bmatrix} \begin{bmatrix} +\text{voc} \\ +\text{cons} \\ -\text{high} \\ -\text{back} \\ -\text{ant} \\ +\text{cor} \\ +\text{cont} \\ -\text{nasal} \\ -\text{strid} \end{bmatrix} \begin{bmatrix} -\text{voc} \\ -\text{cons} \\ +\text{high} \\ -\text{back} \\ -\text{ant} \\ -\text{cor} \\ +\text{cont} \\ -\text{nasal} \\ -\text{strid} \end{bmatrix} \begin{bmatrix} +\text{voc} \\ -\text{cons} \\ -\text{high} \\ +\text{back} \\ -\text{ant} \\ -\text{cor} \\ +\text{cont} \\ -\text{nasal} \\ -\text{strid} \end{bmatrix} \right\}$$

Featural complexity

The disjunctive rule i → y / ___ {p,y,r,a} (simplified)

(17)

$$\begin{bmatrix} -\text{cons} \\ +\text{high} \end{bmatrix} \rightarrow [-\text{voc}] \;/\; \begin{bmatrix} \\ -\text{back} \end{bmatrix} \left\{ \begin{bmatrix} -\text{voc} \\ +\text{cons} \\ +\text{ant} \\ -\text{cor} \\ -\text{nasal} \end{bmatrix}, \begin{bmatrix} +\text{voc} \\ +\text{cons} \\ -\text{ant} \end{bmatrix}, \begin{bmatrix} -\text{voc} \\ -\text{cons} \\ -\text{back} \end{bmatrix}, \begin{bmatrix} +\text{voc} \\ -\text{cons} \\ -\text{high} \\ +\text{back} \end{bmatrix} \right\}$$

Featural complexity

Compare a different rule: i → y / ___ {i,u,æ,a}

- By assumption, this is all the vowels of the language
- The simplest reduction of this rule is…

$$(24) \qquad \begin{bmatrix} -\text{cons} \\ +\text{high} \\ -\text{back} \end{bmatrix} \rightarrow [-\text{voc}] \; / \; \underline{\quad} \begin{bmatrix} +\text{voc} \\ -\text{cons} \end{bmatrix}$$

- Much simpler than the /___ {p,y,r,a} rule on the previous slide

Learning based on grammar length: MDL

Favoring short grammars

- The basic idea of the SPE evaluation metric: favor short grammars
- This alone is not quite right: the shortest grammar is always the empty grammar!
- Chomsky and Halle assumed that the learner was confronted with alternations (e.g., [rad] ∼ [rat]) and forced to capture them with phonological rules
- It’s not always so clear what to include in the phonological grammar or not

The basic problem: what do learners learn?

nasa	bor	itu	rek	wijki
ubi	lig	tev	kez	udag
pim	prog	gojbog	ile	zuf
jer	ulse	uwtu	iwna	arfe
mirpumkemok	anvo	mikod	afavu	dol
irledava	rus	wengja	teni	eswitum
kerid	zep	ewid	vidob	ili
plu	obmuta	urjak	ejje	udi
iji	ris	sujnu	zoj	esnuk
udoldas	bav	wortik	kinwirgafu	zoj
vempiw	wis	dogo	ufo	olwaz
gib	ujoj	dawbmu	muganuz	rupal
glusnum	tawab	poj	fizi	awe
liniw	wis	tile	garap	jazru
tutegnig	ore	fis	mejenlo	zibmo
ulu	top	opin	gew	levmit
dowtug	druf	ruketa	pov	firig

A simpler example: *ab*-nese

Rasin & Katzir (2016), example (1)

(1) *bab*, *aabab*, *ab*, *baab*, *babaaaa*, *babababababaabab*, *aaab*, *bababababaa*, *babaaaa*, *aaab*, *babababababaabab*, *baab*, *bab*, *ab*, *aabab*, *aabab*, *baab*, *babababababaabab*, *aaab*, *bababababaa*, *ab*, *babaaaa*, *bab*, *aaab*, *ab*, *aaab*, *aabab*, *bababababaa*, *baab*

- One response: learn a lexicon
  - List: *ab*, *aaab*, *aabab*, *bab*, etc.
- Another response: learn a grammar
  - Words must have the shape *ab*, *aaab*, *aabab*, *bab*, etc.

Overfitting

- Humans generalize (productivity)
  - Blick tests, wug tests, neologisms, etc.
- Generalization through abstraction
  - General enough to cover not just existing words, but types of words (substrings)
  - Classes of sounds (features)
- “Explanation” = concentrate probability on the kinds of words that actually occur

Abstraction isn’t enough

- We don’t just speak *like* the target language, we speak the actual target language
  - i.e., actual words
- Lexicon: record of actual words, link to meaning

The fitting problem

- Lexicon and grammar operate jointly to fit the data
  - Grammar makes attested strings more likely, but doesn’t actually predict which ones occur and which do not
  - Lexicon determines which words we actually hear, but the existence of one word doesn’t make the existence of another word more likely
- How do we determine which facts the grammar is responsible for explaining?

Back to *ab*-nese

(1) *bab*, *aabab*, *ab*, *baab*, *babaaaa*, *bababababababaabab*, *aaab*, *bababababaa*, *babaaaa*, *aaab*, *bababababababaabab*, *baab*, *bab*, *ab*, *aabab*, *aabab*, *baab*, *bababababababaabab*, *aaab*, *bababababaa*, *ab*, *babaaaa*, *bab*, *aaab*, *ab*, *aaab*, *aabab*, *bababababaa*, *baab*

- Words are composed of letters
- Words are composed of letters of the Latin alphabet
- Words are composed of the letters *a* and *b*
- Words are composed of the letters *a* and *b*, subject to the condition that there are no adjacent *b*'s
- Words are composed of the letters *a* and *b*, subject to the condition that there are no adjacent *b*'s, and the number of *b*'s is a power of 2
- Words are composed of the letters *a* and *b*, subject to the condition that there are no adjacent *b*'s, and the number of *b*'s, is a power of 2, and the number of *a*'s is a Fibonacci number

Finding the right balance

- Words are composed of letters
  - Simple, but insufficiently restrictive
- Words are composed of the letters *a* and *b*, subject to the condition that there are no adjacent *b*'s, and the number of *b*'s, is a power of 2, and the number of *a*'s is a Fibonacci number
  - Fairly restrictive, but quite complex
- Complexity should 'pay off' in restrictiveness

Making complexity pay off

- Adding conditions to the grammar eliminates non-existing strings
- This makes existing strings less coincidental
- Two ways to measure this pay-off
  - Likelihood (Bayesian models)
  - Encoding lengths (MDL)

Restrictiveness and encoding length

- Restrictive grammars allow fewer outputs
- The fewer the outputs, the less information is needed to encode a text (the data)

Compact encoding of texts

A text

the farmer in the dell the farmer in the dell hi ho the derry oh the farmer in the dell

- 8 unique words (*the*, *farmer*, *in*, *dell*, *hi*, *ho*, *derry*, *oh*)
- One possible encoding:

the	000	hi	100
farmer	001	ho	101
in	010	derry	110
dell	011	oh	111

- Text:

  000 001 010 000 011 000 001 010 000 011 100 101 000 110 111 000 001 010 000 011

Also 3 bits

Beyoncé: Pray you catch me (2016)

Prayin’ to catch you whispering I’m prayin’ you catch me listening I’m prayin’ to catch you whispering I’m prayin’ you catch me I’m prayin’ to catch you whispering I’m prayin’ you catch me listening I’m prayin’ you catch me

- Also 8 unique words (3 bits per word)

4 bits

Da Ponte: Largo al factotum (1786)

Figaro! Figaro! Figaro!

Figaro! Figaro! Figaro!

Figaro! Figaro! Figaro!

Ahime, ahime, che furia!

Ahime, che folla!

Uno alla volta, per carità! per carità! per carità!

Uno alla volta, Uno alla volta, Uno alla volta, per carità!

Ehi, Figaro! Son quà. Ehi, Figaro! Son quà.

Figaro quà, Figaro là, Figaro quà, Figaro là,

Figaro su, Figaro giù, Figaro su, Figaro giù...

- 16 unique words: 4 bits with equal length coding

Also 4 bits

Dua Lipa ‘Don’t start now’ (2019)

Oh, oh

Don’t come out, out, out

Don’t show up, up, up

Don’t start now (oh)

Oh, oh

Don’t come out, out

I’m not where you left me at all

- 16 unique words: 4 bits with equal length coding

And another…

Lady Gaga and Ariana Grande ‘Rain on me’ (2020)

I’d rather be dry, but at least I’m alive
Rain on me, rain, rain
Rain on me, rain, rain
I’d rather be dry, but at least I’m alive
Rain on me, rain, rain
Rain on me
Rain on me
Mmm, oh yeah, baby
Rain on me

- 16 unique words: 4 bits with equal length coding
- What is theoretical minimum avg. length per word, given this frequency distribution? (i.e., entropy)

Bit lengths of encodings

- Rasin and Katzir (2016) (consistent with much work on Minimum Description Length), talk about description lengths under the assumption that all words are encoded with the same number of bits
- Not the theoretically shortest possible encoding
- When some words are more frequent than others, this can alter how cheap or costly it is to transmit them (Shannon: Entropy)

Entropy

Entropy (Shannon 1948)

Entropy (Shannon 1948)

$$H = -\sum_{i=1}^{n} p(i)\ log\ p(i)$$

- Measure of uncertainty, or unpredictability of a set of events
- Events = series of die tosses, words in a text, etc.
- Unpredictability measured by how much information you have to transmit in order to convey the data
  - If everyone's on the same page and your language is sensible, things that are highly predictable don't need much spelling out
  - Things that are surprising may need more elaboration

Optimizing encoding

Some words more frequent than others

the	$\frac{7}{20}$	hi	$\frac{1}{20}$
farmer	$\frac{3}{20}$	ho	$\frac{1}{20}$
in	$\frac{3}{20}$	derry	$\frac{1}{20}$
dell	$\frac{3}{20}$	oh	$\frac{1}{20}$

- Give more frequent words shorter encodings (e.g., 0), rarer words longer encodings (e.g., 110)
- Entropy H
  $= -\frac{7}{20} log_2 \frac{7}{20} - 3(\frac{3}{20} log_2 \frac{3}{20}) - 4(\frac{1}{20} log_2 \frac{1}{20}) = 2.626$

Optimizing encoding (*cont.*)

- Given optimal encoding in which more frequent items are shorter, words in this text require average 2.626 bits
  - Modest savings over 3, which we got when all words were equally frequent and we used a equal-length encoding scheme
- The more predictable things are, the less we have to say about them $\rightarrow$ smaller entropy; more uniform distribution $\rightarrow$ higher the entropy

Optimizing encoding

catch	$\frac{7}{39}$	me	$\frac{4}{39}$
you	$\frac{7}{39}$	to	$\frac{3}{39}$
prayin’	$\frac{7}{39}$	whispering	$\frac{3}{39}$
I’m	$\frac{6}{39}$	listening	$\frac{2}{39}$

- Entropy H $= -3(\frac{7}{39} log_2 \frac{7}{39}) - \frac{6}{39} log_2 \frac{6}{39} - \frac{4}{39} log_2 \frac{4}{39} - 2(\frac{3}{39} log_2 \frac{3}{39}) - \frac{2}{39} log_2 \frac{2}{39}$
  ≈ 2.8758

Balancing grammar complexity and restrictiveness

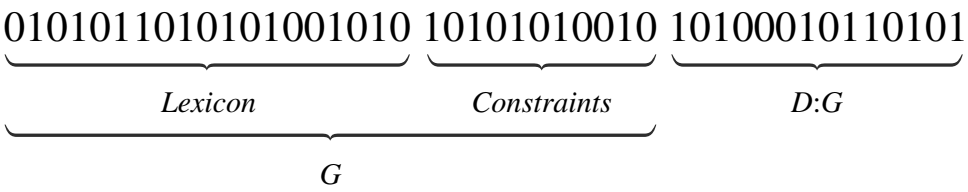

Figure 1
Schematic view of Solomonoff's (1960, 1964) evaluation metric as applied to OT. The grammar *G* consists of both lexicon and constraints. (The bit string in this figure is notional and is only intended as a schematic illustration of how some *G* can be represented using the guidelines discussed in the present section; concrete examples are discussed in detail in section 3.) The data *D* are represented not directly but as encoded by *G*. The overall description of the data is the combination of *G* and *D*:*G*.

- "Grammar" = lexicon + constraints
- Input data, recoded under the grammar = D:G (cf. examples above in bit reduction with a lexicon)

MDL segmentation

Brent & Cartwright, Baroni, Goldsmith

An actual example of child-directed speech

whatareyoudoingsilly?
noyoudontplaywiththestovedillon.
nodillon.
no.
youknowbetterthanthat.
up.
what?
itsadishwasher.
dishwasher.
yeah.
what?
ohreally?
isthatso?

yougotmyslipper?
what?
youreknockingmeover.
iseeyou.
hi.
holdon!
imalmostdone.
ijustgottarinseoffthebottlecleanoffthecountertop andthetableandwellbealldone.
yeah.
ihopeweforgotthatpanoverthere.
alrightigottadoapantoo.
yeah.
idontliketodothemeitherdillon.

A toy example

A short child-directed text

doyouseethekitty;
seethekitty;
doyoulikethekitty;

Analysis 1: long words, simple parses

Lexicon		Segmentation
1	doyouseethekitty	1
2	seethekitty	2
3	doyoulikethekitty	3

A toy example

A short child-directed text

doyouseethekitty;
seethekitty;
doyoulikethekitty;

Analysis 2: simple words, complex parses

Lexicon									
1	d	4	u	7	t	10	i		
2	o	5	s	8	h	11	l		
3	y	6	e	9	k				

Segmentation
1 2 3 2 4 5 6 6 7 8 6 9 10 7 7 3
5 6 6 7 8 6 9 10 7 7 3
1 2 3 2 4 11 10 9 6 7 8 6 9 10 7 7 3

Comparison in terms of description length

- Analysis 1:
  - Lexicon: 47 symbols
  - Parses: 3 symbols
  - Total description length: 50
- Analysis 2:
  - Lexicon: 24 symbols
  - Parses: 49 symbols
  - Total description length: 73 symbols

Another possible analysis

- Analysis 3

Lexicon				Segmentation
1	do	4	the	1 2 3 4 6
2	you	5	like	3 4 6
3	see	6	kitty	1 2 5 4 6

Description Length: $26 + 13 = 39$

- Analysis 4

Lexicon				Segmentation
1	do	4	thekitty	1 2 3 4
2	you	5	like	3 4
3	see			1 2 5 4

Description Length: $25 + 10 = 35$

Back to phonology

Balancing “abstraction” and the lexicon

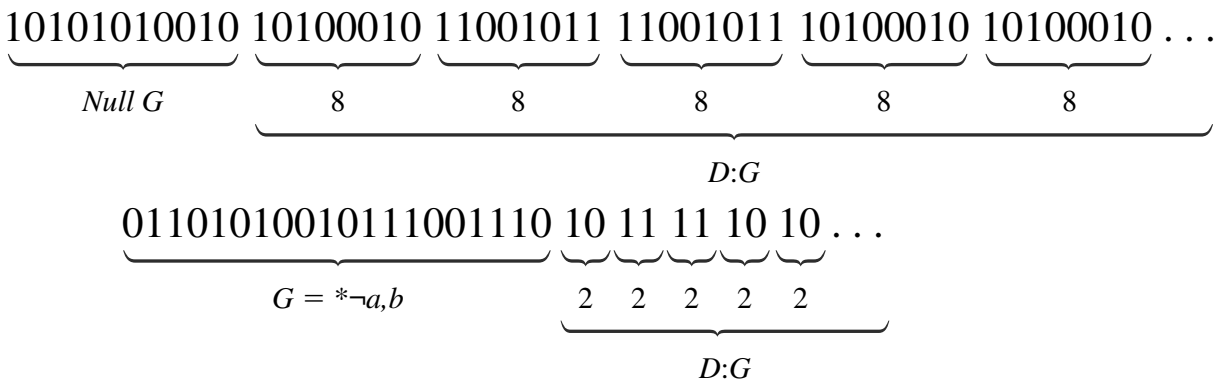

Figure 2
Two simple hypotheses (schematic). The null hypothesis (top) treats the data as an arbitrary sequence of segments. Encoding the grammar is simple, but the price paid for encoding the data is high: eight bits per segment. The hypothesis that treats the data as an arbitrary sequence of *a*’s, *b*’s, and commas requires a slightly more complex grammar, but the savings in encoding the data are noticeable: we now have to pay only two bits per segment.

- Constraint limiting inventory to {’a’, ’b’, ’,’} allows shorter coding of lexicon

Balancing “abstraction” and the lexicon

- Markedness constraint: **bb*
  - Epenthesis: **bb*, MAX ≫ DEP
- Adding one constraint allows us to shorten the encoding of the lexicon by a certain amount

bab	/bb/
aabab	/aabb/
ab	/ab/
bababababababaabab	/bbbbbbbaabb/

The more complex grammar

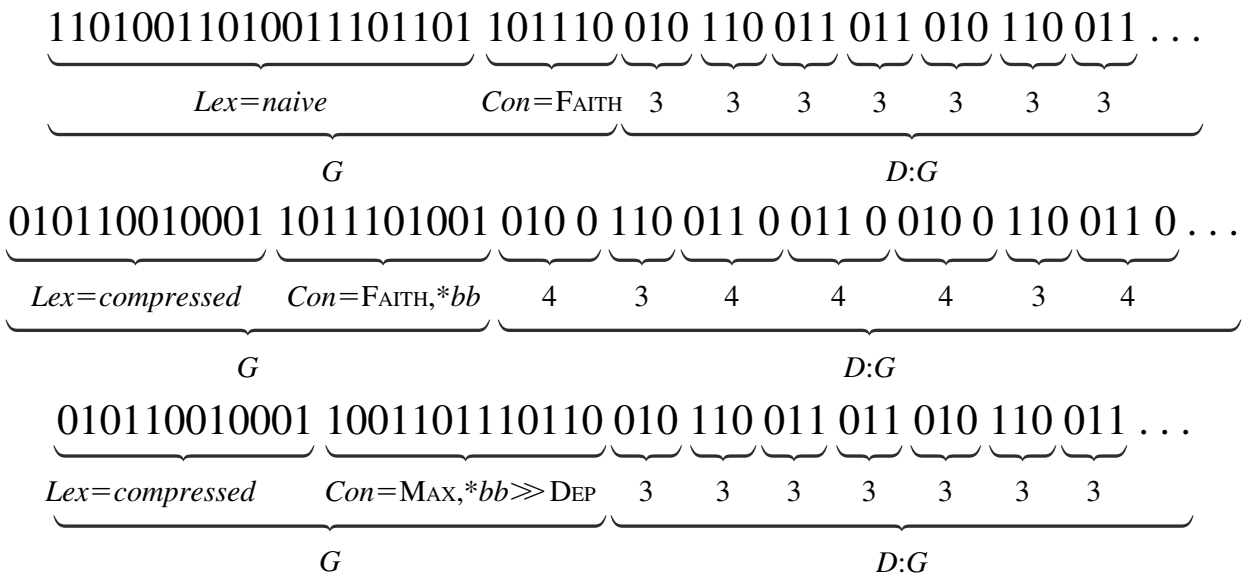

Figure 3
Three more-advanced hypotheses. Introducing a naive lexicon, in which the attested strings are listed, allows us to describe the data word by word rather than segment by segment, yielding significant savings (top). Squeezing the pattern **bb* out of the lexicon results in a shorter lexicon but longer overall description length: for each UR that includes the sequence *bb*, we will now need to specify that the surface form is the result of *a*-epenthesis rather than *b*-deletion (middle). Splitting FAITH into MAX and DEP allows us to maintain both a short lexicon and a short description of the data at the modest cost of a slight complication of the constraints, leading to the shortest overall length (bottom).

Summarizing: balancing the grammar and the data

- Economy of the grammar would favor short and unrestrictive grammars, with long texts
- Economy of representing the data (restrictiveness) favors overfitting, fails to generalize
- Trade-off: minimizing both simultaneously favors capturing “significant” generalizations

Balancing “abstraction” and the lexicon

- Perhaps a bit trickier to use “number of *b*’s must be a power of 2 to shorten the lexicon
  - A plausible repair: insert just enough *b*’s to reach the next power of 2)
- But where to insert them? E.g., *babababaa* could be /abbbaa/, but should that be pronounced [babababaa]? [abababababaa]? [abababababa]? [abababaab]?

Applying this to OT: lengths of representations

Assume that segments are represented as feature matrices

	cons	cont
a	−	+
b	+	−
s	+	+

Lexicon:

(12) a. {*asa, ba, bsab*}

b. − + + + − + # + − − + # + − + + − + + − ##

c. 01000000010010000101001000010000010000011010

Length of representing constraints

- Limited inventory of constraint types
  - *$F_1F_2...F_n$
  - DEP(F)
  - MAX(F)
  - IDENT(F)
- Coding of the constraints

(13) a. DEP(−cons) ≫ MAX(+cont) ≫ *[+cons][−cons, +cont] ≫ IDENT(−cont)

b. D − cons#M + cont#P + cons# − cons + cont##I − cont##

- Lengths

Symbol	Code
D	0000
M	0001
I	0010
P	0011

Symbol	Code
cons	0100
cont	0101

Symbol	Code
+	0110
−	0111

Symbol	Code
#	1000

Figure 5
Binary code assigned to each symbol

Recoding the data (D:G)

- p. 251 “Describing a surface representation that can be parsed by the grammar amounts to specifying two successive choices: a choice of a UR u_i from the lexicon and a choice of an optimal output $o_{i,j}$ of that UR. We assign each choice from the lexicon a fixed binary code as illustrated in (14a) (for the case $m = 5$). Choices from sets of optimal output candidates receive similar treatment (14b): given a UR, all optimal candidates are enumerated; the number of bits required to specify a choice of an optimal candidate depends on the total number of optimal candidates for the UR (in the middle table, no code is needed as the choice is deterministic).”

Recoding the data (D:G) (*cont.*)

(14) a.

UR	Code
u_1	000
u_2	001
u_3	010
u_4	011
u_5	100

b.

u_1	
Output	Code
$o_{1,1}$	00
$o_{1,2}$	01
$o_{1,3}$	10

u_2	
Output	Code
$o_{2,1}$	

u_3	
Output	Code
$o_{3,1}$	0
$o_{3,2}$	1

. . .

Recoding the data (D:G) (*cont.*)

- "Suppose now that we wish to encode s_1 given G. If s_1 cannot be parsed by G, there is no finite binary string that can serve as a description of s1, and its description length will be taken to be infinite. Alternatively, suppose that s_1 is equal to the output $o_{1,3}$ in our example (14b). In that case, s_1 can be described by the binary string 00010 (000 specifies the choice of u_1, 10 the choice of $o_{1,3}$), so its description length is 5. In general, phonological grammars are ambiguous, and it is possible that a given surface representation has more than one parse. For example, s_1 could also be equal to $o_{3,1}$, an output of u_3 under G. When multiple descriptions are available, the shortest one will be chosen. In our

Recoding the data (D:G) (*cont.*)

example, the string 0100 ends up as the shortest description of s_1, a description of length 4.”

Testing this out

- Initial state
  - Single FAITH constraint
  - IN=OUT lexicon
- Search: simulated annealing
  - Add/remove/change a segment from the lexicon
  - Add a single feature bundle constraint to the lexicon
  - Add/remove a feature bundle of a constraint
  - Demote a constraint

Test 1: *ab*-nese

Our first dataset is a language similar to *ab*-nese, presented in section 2.1 and repeated here.

(16) *bab*, *aabab*, *ab*, *baab*, *babaaaa*, *bababababababaabab*, *aaab*, *bababababaa*, *babaaaa*, *aaab*, *bababababababaabab*, *baab*, *bab*, *ab*, *aabab*, *aabab*, *baab*, *bababababababaabab*, *aaab*, *bababababaa*, *ab*, *babaaaa*, *bab*, *aaab*, *ab*, *aaab*, *aabab*, *bababababaa*, *baab*

Given an alphabet $\Sigma = \{a, b\}$ and one feature $\pm$cons ($a = [-\text{cons}]$, $b = [+\text{cons}]$), we generated an initial pool of words by taking all combinations of 1–6 syllables from the set {*a*, *ab*, *ba*, *bab*}. We then filtered out all words that included the sequence *bb* and provided the learner with the resulting set of words ($n = 252$). The full input for this simulation (and the following

(17) *Initial grammar*

$$G_{initial} = \begin{cases} \textsc{Lex}: \ bab, aabab, ab, baab, babaaa, bababababaa, \ldots \\ \textsc{Con}: \textsc{Faith} \end{cases}$$

Description length: $|G_{initial}| + |D{:}G_{initial}| = 4{,}622 + 201{,}600 = 206{,}222$

(18) *Final grammar*

$$G_{final} = \begin{cases} \textsc{Lex}: \ bb, aabb, ab, baab, bbaaa, bbbbaa, \ldots \\ \textsc{Con}: \textsc{Max}([+\text{cons}]) \gg *[+\text{cons}][+\text{cons}] \gg \textsc{Faith} \end{cases}$$

Description length: $|G_{final}| + |D{:}G_{final}| = 4{,}028 + 201{,}600 = 205{,}628$

Test 2: allophonic aspiration

- Stops must be aspirated before vowels, but not elsewhere
- Goal: favor a grammar in which aspiration is removed from the lexicon, because it's predictable
- Implementational choice: aspiration is a segment [ʰ], making the cost of including it easy to calculate directly

(19) a. {k^hat, ip, k^hatpit}

b. $\underbrace{{}^haikpt\#}_{\textit{inventory}}\ \underbrace{k^hat\#ip\#k^hatpit\#}_{\textit{lexicon}}$

(20)

	cons	stop	spread glottis	velar	labial	high
a	−	−	−	−	−	−
i	−	−	−	−	−	+
u	−	−	−	−	+	+
p	+	+	−	−	+	−
t	+	+	−	−	−	−
k	+	+	−	+	−	+
h	+	−	+	−	−	−

Results for aspiration

(21) a. *Initial grammar*

$$G_{initial} = \begin{cases} \text{LEX: } \{a, i, u, p, t, k, {}^{h}\}; up, t^{h}i, k^{h}at, ip^{h}uk, p^{h}ikp^{h}u, t^{h}ik^{h}ut, \ldots \\ \text{CON: FAITH} \end{cases}$$

Description length: $|G_{initial}| + |D{:}G_{initial}| = 4{,}359 + 160{,}000 = 164{,}359$

b. *Final grammar*

$$G_{final} = \begin{cases} \text{LEX: } \{a, i, u, p, t, k\}; up, ti, kat, ipuk, pikpu, tikut, \ldots \\ \text{CON: } *[+\text{stop}][-\text{cons}] \gg \text{FAITH} \gg \text{MAX}([-\text{spread glottis}]) \end{cases}$$

Description length: $|G_{final}| + |D{:}G_{final}| = 3{,}402 + 160{,}000 = 163{,}402$

Test 3: Optionality

- Dell (1981): French *table* pronounced [tab] ∼ [tabl], but *parle* only [parl] not *[par]
- Learning task: learn a lexicon with final CC clusters, and an optional rule of deleting C2 in rising sonority clusters
- Challenge: why not a simpler rule, optionally deleting C2 in any CC cluster?

Test 3: Optionality

- Rasin and Katzir test a similar example, easier to model for technical reasons: /tabl/ pronounced variably as [tab] ~ [tabil]
- Learning challenge: reduce [tab] ~ [tabil] to /tabl/, but non-alternating [paril] not reduced to /parl/
- MDL insight
  - Collapsing [tab] ~ [tabil] to /tabl/ provides significant savings (1 UR instead of 2), while reducing non-alternating [paril] to /parl/ is much smaller savings

Results for optionality

(25) a. *Initial grammar*

$$G_{initial} = \begin{cases} \textsc{Lex}\text{: } tabil, tab, paril, tapil, tap, radil, labil, lab \\ \textsc{Con}\text{: } \textsc{Faith} \gg \textsc{Dep}([-\text{high}]) \gg \textsc{Max}([-\text{liquid}]) \gg *[+\text{cons}] \end{cases}$$

Description length: $|G_{initial}| + |D{:}G_{initial}| = 589 + 600 = 1{,}189$

b. *Final grammar*

$$G_{final} = \begin{cases} \textsc{Lex}\text{: } tabl, paril, tapl, radil, labl \\ \textsc{Con}\text{: } *[+\text{cons}][+\text{cons}] \gg \textsc{Faith} \gg \textsc{Dep}([-\text{high}]) \gg \textsc{Max}([- \end{cases}$$

Description length: $|G_{final}| + |D{:}G_{final}| = 415 + 750 = 1{,}165$

What about actual French?

- Dell's comparison /tabl/ $\rightarrow$ [tab], but /parl/ $\not\rightarrow$ *[par] was not about favoring different UR's
- Here, the problem is to learn to restrict the markedness constraint to a specific *[$-$son][$+$cons] constraint

Test 4: Alternations

- Regressive voicing assimilation

(28) $$G = \begin{cases} \textsc{Lex}: & katav_{\{\text{-}t\}}, daag_{\{\text{-}t\}}; \text{Suffixes:} \{t\} \\ \textsc{Con}: & \text{assimilation-enforcing constraint ranking} \end{cases}$$

(29)

1. katav	3. daag	5. rakad	7. takaf
2. kataft	4. daakt	6. rakadet	8. takaft

- MDL insight: learning the constraints and ranking to produce alternations allows fewer listed allomorphs

Results for alternations

(31) a. *Initial grammar*

$$G_{initial} = \begin{cases} \text{Lex: } katav, daag, rakad, takaf, kataft, daakt, rakadet, \\ \quad takaft; \text{ Suffixes:}\{\ \} \\ \text{Con: Faith} \gg \text{Max([+cons])} \gg \text{Dep([−ATR])} \gg \text{Ident([−voice])} \\ \quad \gg \text{Ident([+cons])} \gg \text{Ident([+labial])} \gg \text{Ident([−labial])} \\ \quad \gg \text{Ident([−high])} \gg \text{Ident([+high])} \gg *[-\text{coronal}][+\text{ATR}] \\ \quad \gg *[+\text{coronal}][+\text{coronal}] \gg *[+\text{cons}, +\text{voice}][-\text{voice}] \end{cases}$$

Description length: $|G_{initial}| + |D{:}G_{initial}| = 864 + 24 = 888$

b. *Final grammar*

$$G_{final} = \begin{cases} \text{Lex: } katav_{\{-t\}}, daag_{\{-t\}}, rakad_{\{-t\}}, takaf_{\{-t\}}; \text{ Suffixes:}\{t\} \\ \text{Con: } *[+\text{cons}, +\text{voice}][-\text{voice}] \gg *[-\text{coronal}][+\text{ATR}] \\ \quad \gg *[+\text{coronal}][+\text{coronal}] \gg \text{Ident([−high])} \gg \text{Ident([−voice])} \\ \quad \gg \text{Dep([−ATR])} \gg \text{Faith} \gg \text{Ident([+labial])} \gg \text{Max([+cons])} \\ \quad \gg \text{Ident([−labial])} \gg \text{Ident([+cons])} \gg \text{Ident([+high])} \end{cases}$$

Description length: $|G_{final}| + |D{:}G_{final}| = 520 + 16 = 536$

From simplicity to markedness

“Some unresolved problems”

Chomsky & Halle (1968), chapter 9

- Voiced obstruents frequently pattern together in phonological processes, but voiced segments rarely do
  - Yet [+voi] is simpler than [+voi,−son]
- Rule 1 involves more featural changes than rule 2:
  - k $\rightarrow$ tʃ / ___ [−back]
  - p $\rightarrow$ t / ___ [−back]

  …Yet the former is common, and the latter is rare or unattested

Markedness

Chomsky and Halle's conclusion (p. 402)

All of these examples, and many others like them, point to the need for an extension the theory to accommodate the effects of the intrinsic content of features, to distinguish expected or natural cases of rules and symbol configurations from others which are unexpected and unnatural. In the linguistically significant sense of the notion "complexity," a rule that voices vowels should not add to the complexity of a grammar but a rule that unvoices vowels should, whereas in the case of obstruents the opposite decision is called for.

Markedness

- “Classical OT” directly answers the call to make “unmarked” outcomes free, by providing innate Markedness constraints
  - However, no way to add markedness-increasing processes, at any cost
- Another possibility: constraint induction, guided by various factors
  - Phonetic grounding (Hayes & Steriade, 2004)
  - Complexity (Heinz et al)

References

Chomsky, Noam, and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper and Row.

Hayes, Bruce, and Donca Steriade. 2004. The phonetic basis of phonological markedness. *Phonetically based phonology*, ed. by Bruce Hayes, Robert Kirchner, and Donca Steriade, 1–33. Cambridge: Cambridge University Press.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41.623–661. Online: http://www.jstor.org/stable/40926398.

Rasin, Ezer, and Roni Katzir. 2016. On evaluation metrics in Optimality Theory. *Linguistic Inquiry* 47.235–282.