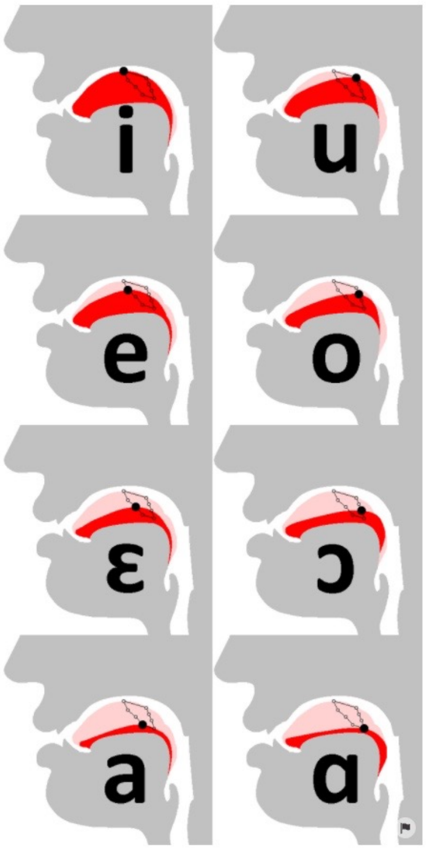

24.901/931, 2026
Introduction to Phonology

Lecture 14

This week

- Midterm Thursday. Materials posted.
- Questions about the midterm?
  I can meet people today 1-2, Wednesday before 2pm. Let me know.
  I can Zoom at other times. We can email. Will post Qs and As.
- Read: McCarthy 2008 Chapter 2.
- Optional: Kager 1999 chapters 1-2.
- **Problem set** 4, an OT exercise, is posted; due next Tuesday.

Today

- Recap basics of OT
- Translate rule-based analyses into OT-analyses.

Recap

- Rule-based phonology is sometimes helped by 2 constraint types:
  - (a) constraints on SR sequences; in OT, *Markedness* (M) constraints
  - (b) preservation constraints; in OT *Faithfulness* (F) constraints
- Optimality Theory grammar: reduced to ranked M and F constraints.

  No rules, no constraints demanding changes in input, no rule order.
- To turn a set of constraints into an input-output mapping procedure, constraints must be ranked.
- One must establish the alternatives (*candidate SRs*) that the grammar chooses from.

An old slide on M and F in Turkish compounds

UR çam, çevi	Preserve 1st root V	Harmony	Comment
☞ çam#çevi	√	*(aC_0e)	Satisfies top constraint; violates minimally the lower
çam#çavɨ	* (e⟶a)	√	Violates higher constraint
çem#çevi	* (a⟶e)	√	Violates higher constraint

2 constraints, ranked; being leftmost signals ‘has priority’
3 main ways to realize this compound: 3 SR candidates

More conflict: Turkish [o,ø]*

*Turkish words have no o,ø except in 1st syll. Loking to explain that 'except' part.
Property of vowel inventories: u, y exists but no o, ø, ɔ, ɒ; never the converse.

- Inventory constraint (M): No [+round, -high] V (no [o, ø])

☞	UR odA-lAr	Preserve 1st root V	Inventory	Harmony
	odalar	√	*(o)	* (oda)
	odolor	√	***	√
	odolar	√	**	* (dola)
	adalar	*(o→a)	√	√

Elements of OT phonology

Constraints

- Functions from classes of linguistic structures to violation marks (*).

  E.g. a *HH constraint assigns 1 * to each sequence of adjacent Hs, HH.

- A form that violates a constraint C is assigned one * for *each instance* it contains of the structure that C penalizes.

- ba da ga mi lo how many violations assigned by *HH?

```
ba   da   ga   mi   lo
 |    |    |    |    |
 H    H    H    L    H
```

Markedness constraints (M*)

- Evaluate properties of SRs
  - a. *[+strident]-[+strident]
  - b. *[α round][-α round]
- UR not consulted in evaluating M constraints
  - e.g. SR [bʌsəz] < UR /bʌsz/ doesn't get a * for *strident-strident.
  - only an SR [bʌsz] (from any UR) does.

*Why call the M constraints *Markedness*? It's a reference to Chomsky and Halle's (1968) *Theory of Markedness*, which distinguishes between structures that are cross-linguistically widespread, preferable(*unmarked*) and dispreferred ones (*marked*). The aim for Chomsky and Halle was to explain why certain structures recur cross-linguistically. The Theory of Markedness is not in use now but its objectives and the term remains.

Faithfulness Constraints (*Faith*, *F*, or *Correspondence*)

- Faith constraints evaluate how faithfully each SR preserves its UR.
  Each mismatch is penalized.
- Example 1: a constraint against any SR that loses a UR tone.
  MAX TONE: a violation for each UR tone that's absent from SR
- Example 2: constraint against any SR that **loses** some UR segment type
  MAX V: a * for each UR V that's absent from SR.
- Example 3: constraint against any SR containing a sound **absent** from UR
  DEP V(or C): a * for each SR V(C) that's absent from any UR
- Example 4: constraint against any SR containing a **modified** sound vs. UR
  IDENT F: a * for each SR segment whose F value differs from its UR F value

Why MAX, why DEP? UR must be **MAX**imally preserved in SR, SR is maximally **DEP**endent upon the UR

The M-F division and phonological universals

- The same M constraints occur across languages. Some are likely universal.
- Phonological systems differ in extent to which these M are enforced:

  e.g. Margi, Kinyarwanda strictly enforce *VV;

  English does not: *real*, *koala; thirtyish* (still, VV seqs are very rare)
- A dilemma:
  - preserve UR unchanged, tolerating some M violations in SRs?
  - eliminate M violations, modifying the UR and violating F?
- Tolerate: F >> M　　　Modify: M >> F
- Generalize: difference betw. phonological systems is ranking of constraints

Why rank? Conflict and violability

- Constraints conflict w. other constraints, esp. when stated most generally.
  E.g. the constraint that causes VH conflicts with preservation of 1st V.
- C_1 conflicts with C_2 = not possible to satisfy C_1 without violating C_2 in some forms

	/çam-çevi/	**Don't change root V1**	***[α back][-α back]**
	çam-çavɨ	* (çev →çav)	
	çem-cevi	* (çam→çem)	
☞	çam-çevi		* (çamçevi)

More complex constraints can replace ranking: e.g. ***[α back][-α back]** ***inside 1 word.***
That can also exclude VH in compounds.
But ranking allows the more general statement ***[α back][-α back]**.

The grammar reduces to a constraint hierarchy, H

- H = a priority order, or ranking, over the entire constraint set
  $C_1 >> C_2 >> \ldots >> C_{n,}$
  where >> = "outranks, has priority at being satisfied, over _ "

/UR/	C_1	C_2	C_3
SR1	…	…	…
SR2	…	…	…

SR1 = one candidate SR realization of this UR
SR2 = another candidate for same UR
H determines which SR is chosen

- All constraints, M and F, are violable.
- But each violation is assessed a * mark. Marks determine how viable a form is, compared to other candidates.
- All constraints could be ranked; not all rankings are discoverable in the data.

2 functions help map URs into SRs

- *Generation* (**Gen**): turns any UR into a set of potential SRs, or *candidates*.
- *Evaluation* (**Eval**): selects the best candidate according to the language's Hierarchy.
- Short Eval
  The best candidate for a UR violates lower ranked constraint(s) than other candidates.
  Or, if it violates the same constraints as other candidates, it does so less than the others.

A first informal illustration of Gen and Eval

- We consider the case of a learner who knows the URs of some words, their actual SRs, and all constraints, but not yet their rankings.
- The learner discovers rankings by comparing the observed SR with what Eval judges to be the optimal SR (= the *winning candidate*).
- If winning candidate ≠ the observed SR, rankings are modified, or analysis is changed in other ways,.

Example: Margi /hù-árì/ → [hwǎrì]
First, we illustrate what Gen does

The UR

The faithful SR

Some SR candidates

Actual SR

hù -árì **| ||** **L HL**	**H goes here**			
hù -árì | || L HL				
hù -tárì | || L HL				
hw -árì || H L				
hw -ǎrì /\\\| LHL				

How to select candidates to discover rankings

- Lots and lots of candidates. Very few are relevant to discovering rankings.
- Always consider the *faithful* candidate = unmodified UR, to understand *why the UR doesn't surface intact*, if that's the case.
- Always consider SR. To check that the grammar produces the right SR.
- Consider also any other modification of the UR that satisfies the M constraints considered.
- More on this below.

Now we illustrate informally Eval. We know some constraints.

The UR

The faithful candidate

Other SR candidates

The actual SR

<table>
<tr><th>hù -árì
| | |
L HL</th><th>*VV</th><th>Dep C</th><th>Max Tone</th><th>Max V</th></tr>
<tr><td>hù -árì
| | |
L HL</td><td></td><td></td><td></td><td></td></tr>
<tr><td>hù -tárì
| | |
L HL</td><td></td><td></td><td></td><td></td></tr>
<tr><td>hw -árì
| |
H L</td><td></td><td></td><td></td><td></td></tr>
<tr><td>hw -ǎrì
/\ |
LHL</td><td></td><td></td><td></td><td></td></tr>
</table>

Step 1: record constraint violations

The UR

The faithful SR

Some SR candidates

The actual SR

<table>
<tr><th>hù -árì
| | |
L HL</th><th>*VV</th><th>Dep C</th><th>Max Tone</th><th>Max V</th></tr>
<tr><td>hù -árì
| | |
L HL</td><td>* (u-a)</td><td></td><td></td><td></td></tr>
<tr><td>hù -tárì
| | |
L HL</td><td></td><td>* (t)</td><td></td><td></td></tr>
<tr><td>hw -árì
| |
H L</td><td></td><td></td><td>* (L→Ø)</td><td></td></tr>
<tr><td>hw -ǎrì
/\ |
LHL</td><td></td><td></td><td></td><td>(satisfied: u=w)</td></tr>
</table>

Step 2: compare candidates. A winner emerges.

The UR

The faithful SR

Some SR candidates

Perfect!
...given this constraint set

<table>
<tr><th>hù -árì
| ||
L HL</th><th>*VV</th><th>DEP C</th><th>MAX TONE</th><th>MAX V</th></tr>
<tr><td>hù -árì
| ||
L HL</td><td>*</td><td></td><td></td><td></td></tr>
<tr><td>hù -tárì
| ||
L HL</td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>hw -árì
| |
H L</td><td></td><td></td><td>*</td><td></td></tr>
<tr><td>hw -ǎrì
/\|
LHL</td><td></td><td></td><td></td><td>(satisfied: u=w)</td></tr>
</table>

Actually, winner isn't perfect: contour tones are avoided, $*T_1\smile T_2$

The UR

The faithful SR

Some SR candidates

The actual SR

<table>
<tr><th>hù -árì
| | |
L   HL</th><th>*VV</th><th>Dep C</th><th>Max Tone</th><th>Max V</th><th>$*T_1\smile T_2$</th></tr>
<tr><td>hù -árì
| | |
L   HL</td><td>*</td><td></td><td></td><td></td><td></td></tr>
<tr><td>hù -tárì
| | |
L   HL</td><td></td><td>*</td><td></td><td></td><td></td></tr>
<tr><td>hw -árì
| |
H L</td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>hw -ǎrì
/\ |
LHL</td><td></td><td></td><td></td><td>(satisfied: u=w)</td><td>*</td></tr>
</table>

A ranking (= solid line) helps

The UR

hù -árì | | | L HL	*VV	Dep C	Max Tone	Max V	*T_1T_2
		Equally ranked at the top			Lower ranked
hù -árì | | | L HL	*				
hù -tárì | | | L HL		*			
hw -árì | | H L			*		
hw -ǎrì /\ | LHL				(satisfied: u=w)	*

The faithful SR

Some SR candidates

The actual SR. Imperfect, but the best, given this H

Winner violates a 2nd constraint

The UR

Equally ranked at the top, for now

Violated if V→ glide

hù -árì **| | |** **L HL**	***VV**	**Dep C**	**Max Tone**	**Max V**	***T_1‿T_2**	**Ident [±syll]**
hù -árì | | | L HL	*					
hù -tárì | | | L HL		*				
hw -árì | | H L			*			*
hw -ǎrì /\\ | LHL				(satisfied: u=w)	*	* (u→w)

The faithful SR

Some SR candidates

Actual SR. imperfect, but best, given this H

Eval less informally. Presupposes rankings are known.

- Take two candidates, c_x and c_y, and their violation scores for all constraints.
- Find the highest ranked constraint C that differentiates c_x from c_y.
- If c_x violates C fewer times than c_y, c_x is better than c_y ($c_x \succ c_y$)

(i)	C_1	C_2	C_3	C_4
c_y	*			
c_x		*		

(ii)	C_1	C_2	C_3	C_4
c_y	*	*		
c_x	*			

(iii)	C_1	C_2	C_3	C_4
c_y	*	*		
c_x	*		*	*

(iv)	C_1	C_2	C_3	C_4
c_y	* *	*		
c_x	*			

(v)	C_1	C_2	C_3	C_4
c_y		* *		* * *
c_x		* *		* *

After C_y is eliminated, repeat comparison, now between c_x and c_z. Continue until all candidates are compared.

With graphic aids

- LR order of constraints mirrors rankings (highest leftmost), unless separated by dotted lines, which indicate that their ranking is unknown.
- Critical violation (marked *!): the violation causing a candidate to lose to another. Winning candidate is marked by ☞.

(i)	C_1	C_2	C_3	C_4
c_y	*!			
☞ c_x		*		

(ii)	C_1	C_2	C_3	C_4
c_y	*	*!		
☞ c_x	*			

(iii)	C_1	C_2	C_3	C_4
c_y	*	*!		
☞ c_x	*		*	*

(iv)	C_1	C_2	C_3	C_4
c_y	* *!	*		
☞ c_x	*			

(v)	C_1	C_2	C_3	C_4
c_y		* *		* * *!
☞ c_x		* *		* *

Terminology

- The *winning candidate* is the form designated by the grammar (= constraint hierarchy, H) as the best SR for a given UR, given H.
- *Tableau* or *violation tableau*: an evaluation by H of competing SRs for *one* UR
- c_x *better satisfies* some C than c_y: c_x has fewer violations of C than c_y
- Some C *favors* c_x over c_y: c_x has fewer violations of C than c_y

More on Faithfulness

- F constraints assign *s marks when SRs mismatch corresponding URs in specific ways.
- Mismatch types
  1. SR lost a piece of UR; or added a new piece to the UR
  2. SR still has that piece of UR, but has changed it
  3. SR has changed relations between pieces of the UR that it preserved
- Type 1: e.g. MAX SEG/ MAX TONE (= no deletion of segments; of tones)

  One * for each UR segment/tone that lacks a SR correspondent.

  e.g. DEP SEG/ DEP TONE (= no insertion of segments/tones)

  One * for each SR segment/tone that lacks a UR correspondent.
- Type 2: e.g. IDENT ±SYLLABIC; other IDENT, one for each F.
- Type 3: e.g. LINEARITY: don't change precedence relations between segments/tones

  if a>b in UR, and if a and b are preserved in SR, then *b >a in SR

More on *Gen(eration)*

- Gen produces, for any UR, a set of SR candidates for that UR.
- There *could* be restrictions on that set:

  The Gen of 1993 OT produces only SR candidates that differ from UR in constituent structure (e.g. syllables). Not in features.

  The Gen of Harmonic Serialism (OT offshoot) produces candidates that improve the input by changing 1 thing (e.g. insert/delete 1segment; change 1F)
- The Gen of classical OT (1995-) has very few restrictions. Any utterable string can be an SR candidate, for any UR. In practice, few candidates are viable. See below.

Translating a rule-based analysis into OT,

as one way to explore OT analyses of some data pattern

This presupposes you have hypotheses
- about decomposing words into morphemes
- about URs of morphemes
- about processes that map URs to SRs

Rule reminders
Rule: A→B/C_D
Structural Description: CAD
Structural Change: A→B

From rule-based to constraint-based analyses

- Start with *one* rule.
- Assume the rule's SD is the only trigger M, M_{SD}. You may revise this.
- The rule applies. Ergo, M_{SD} is satisfied, sometimes, when conflicting with F. Hence, M_{SD} >> some F.
- The rule's SC corresponds to at least 1 F constraint, F_{SC}: so M_{SD} >> F_{SC}
- Other F constraints likely outrank F_{SC} : M_{SD} (...) F_1...F_n >> F_{SC}

  E.g. Margi Glide Formation: *VV >> …. >> IDENT [±SYLLABIC]

  M_{SD} = *VV, F_{SC} = IDENT [±SYLLABIC]

  MAX V >> IDENT [±SYLLABIC

A familiar pair of English rules

Strident-final root		Voiceless non-strident final		Voiced non-strid. final	
(a) Ø→ə		(b) z→s		(c) no change	
buses	bʌsəz	packs	pæks	bags	bægz
buzes	bʌzəz	cats	kæts	bathes	beɪðz
bushes	bʊʃəz	maps	mæps	lobes	loʊbz
hutches	hʌʧəz	cuffs	kʌfs	lives	laɪvz
badges	bæʤəz	mouths	maʊθs	moans	moʊnz

Affix UR = /z/

Epenthesis: Ø→ə/[+strident]_[+strident]

Voicing Assimilation: [-son] →[α voice]/[α voice, -son]_#

Order: Epenthesis > VAssim (bleeding; transparent)

Translating this into an OT analysis

- For each rule, find at least these 3 things:
    - ➡ the M that causes a change between UR and SR
    - ➡ the F constraint(s) violated to satisfy this M
    - ➡ any other F constraint(s) whose violation *could* have to satisfied this M
- The simplest possibility:
    - ➡ the trigger M = the SD of the rule
    - ➡ there is a unique F constraint, such that the trigger M >> *that* F
    - ➡ F constraint can be read off the rule's SC:

      it's penalizes the change corresponding to the SC
    - ➡ the only ranking needed is M >> F. No other F constraints relevant

Translating English inter-strident epenthesis

- Epenthesis: Ø→ə/[+strident]_[+strident]
- Structural Description (SD): [+strident][+ strident]
- Structural Change (SC): Ø→ə
- SD translates into this M constraint: *[+strident][+ strident]
- SC translates into an F constraint that's a form of DEP (*don't insert x*)
- H, thus far: *[+strident][+ strident] >> DEP ə

  means '*this* M *is* satisfied, at the expense of this F'.
- More rankings needed: other F than DEP could be violated to satisfy this M

  So, we need to add at least this: all other relevant F >> F_{lowest}

  This reflects the choice of change = choice of F to violate in satisfying this M
- *Relevant* F: all F constraints that *could be violated to satisfy this M*. Here: MAX, DEP, some IDENT, LINEARITY. Not all F are always relevant: e.g. MAX T, Ident ±rnd aren't here.

The inter-strident epenthesis rule

Rule
Ø→ə/[+strid]_[+strid]

We break down the rule into SD and SC

Rule
Ø→ə/[+strid]_[+strid]
SD = [+strid] [+strid]
SC = Ø→ə

We turn SD into M, SC into F; and M >> F

Rule	Constraint analysis	
Ø→ə/[+strid]_[+strid]	Constraints	Ranking
SD = [+strid] [+strid]	M = *[+strid] [+strid]	M >> F
SC = Ø→ə	F = DEP ə	

Remember: *XY = assign one * mark for each instance of XY in a SR

Check that analyses are equivalent for an initial data set.

Rule	**Constraint analysis**	
Ø→ə/[+strid]_[+strid]	Constraints	Ranking
SD = [+strid] [+strid]	M = *[+strid] [+strid]	M >> F
SC = Ø→ə	F = DEP ə	

	UR	M	F
	/bʌs-z/	*STRID-STRID	DEP
a. (faithful)	bʌsz	*!	
☞b. (improved)	bʌsəz		*

Next steps

1. Make precise the statement of all constraints invoked.
2. Check rankings between F_{SD} and other F constraints.
3. Look for other candidates. The analysis should predict that *they* don't win.
   Two poss. reasons: (i) they violate higher M; (ii) they violate higher F.
4. Verify **the ranking invoked is right for the rest of grammar**.
   Check interactions with other processes
5. Watch out for possible simplifications in the structure of trigger M, by invoking other reasonable Ms, and ranking those.

Checking other candidates that satisfy the M constraint

Consider 5 that satisfy M by violating MAX, DEP

		M	F	F
	/bʌs-z/	*STRID-STRID	MAX	DEP
a	bʌsz			
b	bʌs			
c	bʌz			
d	bʌsəz			
e	bʌsiz			
f	bʌsəzə			

Other candidates that satisfy the M constraint

Assign violations

		M	F	F
	/bʌs-z/	*STRID-STRID	MAX	DEP
a	bʌsz	*!		
b	bʌs		*	
c	bʌz		*	
d	bʌsəz			*
e	bʌsiz			*
f	bʌsəzə			**

We want (d) to win. Select a ranking with constraints favoring (d) at the top.

Other candidates that satisfy the M constraint

		M	F	F
	/bʌs-z/	*STRID-STRID	MAX	DEP
a	bʌsz	*!		
b	bʌs		*!	
c	bʌz		*!	
d	bʌsəz			*
e	bʌsiz			*
f	bʌsəzə			**!

Other candidates that satisfy the M constraint

		M	F	F
	/bʌs-z/	*STRID-STRID	MAX	DEP
a	bʌsz	*!		
b	bʌs		*!	
c	bʌz		*!	
d ☹	bʌsəz			*
e !☞	bʌsiz			*
f	bʌsəzə			**!

☹ = SR not the (unique) winner

! ungrammatical but emerges as equally good

Refine the F constraint, add a higher F

		M	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	Dep ¬ə	DEP ə
a	bʌsz	*!			
b	bʌs		*!		
c	bʌz		*!		
d ☞	bʌsəz				*
e	bʌsiz			*!	
f	bʌsəzə				**!

Definitions

- DEP ə One * for every ə in SR that doesn't correspond to a UR segment
- DEP ¬ə. One * for every non-ə that doesn't correspond to a UR segment
  (property unique to ə: shortest in English; Dep can refer to this)

/bʌs-z/→ [bʌsəz]

More candidates and more F constraints to exclude them

		M	F		F	F
	/bʌs-z/	*STRID-STRID	MAX		DEP¬ə	DEP ə
a	bʌsz	*!				
b	bʌs		*!			
c	bʌz		*!			
d	sbʌz					
e	bʌsiz				*!	
☞f	bʌsəz					*
g	bʌsəzə					**!

LINEARITY

a. One * for each pair xy in UR, x>y (> = precedes), with SR correspondents, x-a, y-b, where b>a

b. One * for every pair xy in UR, x>y, with correspondents in SR, x-a, y-b, where ¬ (a>b) (a and b may be simultaneous or flipped)

[...and more...]

/bʌs-z/→ [bʌsəz]

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	DEP¬ə	DEP ə
a	bʌsz	*!				
b	bʌs		*!			
c	bʌz		*!			
d	sbʌz			*!* (s-ʌ, s-b)		
e	bʌsiz				*!	
☞f	bʌsəz					*
g	bʌsəzə					**!

Other F constraints referring to relations betw. sounds: CONTIGUITY + others
Here, s was contiguous to ʌ, z in UR, but no longer is in SR.

/bʌs-z/→ [bʌsəz]

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	DEP¬ə	DEP ə
a	bʌsz	*!				
b	bʌs		*!			
c	bʌz		*!			
d	sbʌz			*!* (s-ʌ, s-b)		
e	bʌsiz				*!	
☞f	bʌsəz					*
g	bʌsəzə					**!

Propose a candidate that wins if H contains *only* constraints+rankings here.

/bʌs-z/→ [bʌst]. Propose a fix.

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	DEP¬ə	DEP ə
a	bʌsz	*!				
b	bʌs		*!			
c	bʌz		*!			
d	sbʌz			*!*		
e	bʌsiz				*!	
☹ f	bʌsəz					*!
g	bʌsəzə					**!
! ☞h	bʌst (t<z)					

/bʌs-z/→[bʌsəz]: IDENT ±STRID or ±CONT >>DEP ə

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	ID ±STRID	DEP ə
a	bʌsz	*!				
b	bʌs		*!			
c	bʌz		*!			
d	sbʌz			*!*		
☞e	bʌsəz					*
f	bʌsəzə					**!
g	bʌst (t<z)				*!	

Harmonic bounding and size of candidate sets

Added candidates c and f; consider also e

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	ID ±STRID	DEP ə
0.	bʌs-z	*				
a	bʌs		*!			
b	bʌ-z		*!			
c	bʌ		*!*			
☞d	bʌsə-z					*
e	bʌsə-zə					**!
f	əəbʌsə-zəə					**!***
g	bʌs-t				*!	
h	sbʌ-z			*!		

Which candidates could *win under some change of ranking*?
Which candidates would *never win*? Why?

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	ID ±STRID	DEP ə
0.	bʌs-z	*				
a	bʌs		*!			
b	bʌ-z		*!			
c	bʌ		*!*			
☞d	bʌsə-z					*
e	bʌsə-zə					**!
f	əəbʌsə-zəə					**!***
g	bʌs-t				*	
h	sbʌ-z			*!		

One candidate *harmonically bounds* another

		M	F	F	F	F
	/bʌs-z/	*STRID-STRID	MAX	LINEARITY	ID ±STRID	DEP ə
0.	bʌs-z	*				
a	bʌs		*!			
b	bʌ-z		*!			
c	bʌ		*!*			
☞d	bʌsə-z					*
e	bʌsə-zə					**!
f	əəbʌsə-zəə					**!***
g	bʌs-t					
h	sbʌ-z					

For *this* constraint set, (c) has a superset of the violations of (a,b) – aka is harmonically bounded by (a, b); (e,f) are harmonically bounded by (d).

Harmonic bounding

- Let c_1 and c_2 be candidate SRs corresponding to *the same* UR.
- Let S be a constraint set.
- Def: c_1 harmonically bounds c_2 iff c_1 has a proper subset of c_2's violations of the constraints in S.
- This means: c_2 will not win under any ranking over S.
- Under some rankings over S, c_1 may be the best candidate. Then c_1 will win.
- Under other rankings, another candidate may be better than c_1, so c_1 will lose.
- But c_2 will always have a worse score compared to c_1.
- C_2 will never win **because c_1 is available**, and **it's always a better deal**.

Harmonic bounding and the infinite set

- The candidate set is infinite (in classical OT) because it's mostly populated by infinities of harmonically bounded losers.
- *bʌsəzə, bʌsəzəə, bʌsəzəəə, bʌsəzəəəə, bʌsəzəəəəə, bʌsəzəəəəəə*, ..., all bounded by *bʌsəz*.
- Plus *some very few* candidates with a chance to win under some ranking over S.
- So, the relevant candidates from some UR (= contenders) are
  - i. the fully faithful candidate
  - ii. unfaithful ones that better satisfy *some M constraint violated by the UR* and suffer minimal F violation

(How do we know the relevant M constraints? We don't know *a priori* all M constraints, but we have guesses about factors that qualify a hypothetical constraint as M. Cf. Hayes and Steriade 2004, on Canvas).

Harmonic bounding and the infinite set

- How large is the *set of contenders* for a given UR?
- How many computational steps are needed to generate and evaluate them?
- Discussion of options in Riggle 2011, under OT Basics

How do we find *all* the relevant candidates?

- *Relevant* = relevant for verifying that a hierarchy H selects correct SRs for URs. Suppose the only M is *STRIDENT STRIDENT.
- H = M_{SD} (>>) F_n…>>F_{SC}
- Take candidates relevant to this one M, *STRID-STRID .
- There are 4 ways that *STRID-STRID can be satisfied, given a UR like /sz/:
    - Remove one or both stridents: MAX violated
    - Separate them, insert a buffer: DEP violated; varieties of DEP
    - Move one or both out of the way: LINEARITY violated
    - Change one or both to [-strident]: IDENT [±STRIDENT] violated
- More options?
- I didn't think so.