Computing assignment: Harmonic Serialism

due Thursday, 5 Oct. 2015

Overview and goals

- Learn how to use the OT-Help software to generate typologies in Harmonic Serialism.
- Implement a HS analysis of a phenomenon of your choice

Step-by-step instructions

Download and install OT-Help

1. Follow the instructions at http://people.umass.edu/othelp/
2. Take a look at the manual for the serial version. Read Section 1.

Do a test run using sample files

3. Work through Section 2 of the manual step by step, up through page 33
4. Tip for Windows users: I recommend opening the text files in Notepad++ (https://notepad-plus-plus.org/) rather than Notepad when you want to look at or edit them. Line breaks and tabs will work better.

Run your own HS typology

5. Choose a phenomenon—keep it simple!!
6. Start by making the constraint file. Section 4 of the manual will help you see how to define various types of markedness constraints, including ALIGN, scalar constraints, and long-distance constraints. I'm suggesting starting here because if you find that you can't define the constraints you want, you might want to pick a different phenomenon.
7. Then make the operation file. Remember that this is effectively where you define the faithfulness constraints too. Section 3 of the manual can help, especially if you want to get fancy.
8. Finally, make the input file, and run it.

What to turn in

- See example below for what your write-up should include. Note that it's fine to adapt an analysis we saw in class or that you've read about, or even that you've proposed in a homework in a previous class.
- I'll also provide a spot on CCLE where you can upload your three files (input file, constraints file, operations file).
- But do your best to ensure that I won't look at those files: your write-up should have all the information I'd want.

Example writeup

Introduction, including basic analysis in parallel OT

I chose to implement a schematic version of Central Venetan metaphony. Here are the basic data, taken from Class 10's handout, with piece-by-piece analysis in parallel OT (a simplified version of Walker's analysis, as we discussed in class).

- Stressed {é,ó} raise to {í, ú} when followed by a high suffix vowel:

without a high suffix	*with a high suffix*	
kals-**é**t-o	kals-**í**t-i	'sock (m sg/pl)'
m**ó**v-o	m**ú**v-i	'move (1 sg/2 sg)'

  The driving constraint is LICENSE(high, stressed): the feature [+high] must be associated to a stressed vowel. I assume that in a candidate like [múv-i], the feature [+high] is autosegmentally associated to both vowels, in violation of DON'TSHARE, even though I don't draw the autosegmental representation.

/móv-i/	LICENSE(hi,stress)	DON'TSHARE
móv-i	*!	
→ múv-i		*

- Lax vowels ([ɛ,ɔ,a]) don't raise, because there is no legal vowel in the inventory ([i,e,ɛ,a,u,o,ɔ]) for them to raise to:

g**á**t-o	g**á**t-i	'cat (m sg/pl)'	
v**ɛ**tʃ-o	v**ɛ**tʃ-i	'old man (m sg/pl)'	*vɪ́tʃ-i

  I'll use the constraint *HIGHLAX (no high, lax vowels) to rule out raising in these cases. IDENT(tense) prevents the repair of making the vowel tense.

/ vɛ́tʃ-i /	IDENT(tense)	*HIGHLAX	LICENSE(hi,stress)	DON'TSHARE
→ vɛ́tʃ-i			*	
vɪ́tʃ-i		*!		*
vítʃ-i	*!			*

- [+high] can spread through an intervening unstressed vowel:

órd**e**n-o	**ú**rd**i**n-i	'order (1 sg/2 sg)'	

  I assume that [+high] can't associate to the first and last vowels, skipping a middle [-high] one ([úrden-i]), because there would be crossed association lines—that is, I assume thatthis is not even a candidate.

/ órden-i /	LICENSE(hi,stress)	DON'TSHARE
órden-i	*!	
órdin-i	*!	*
→ úrdin-i		**

- No spreading occurs through an intervening /a/, because /a/ can't raise, as we saw above:

  lav**ór**-**a**-v-a lav**ór**-**a**-v-i 'work (1 sg [3sg?] perf/2 sg impf)'

  (I use "I" for the high, central, unrounded, lax mystery vowel that /a/ would become if it raised)

/ lavór-av-i /	*HIGHLAX	LICENSE(hi,stress)	DON'TSHARE
→ lavór-av-i		*	
lavór-Iv-i	*!	*	*
lavúr-Iv-i	*!		**

- If there is no raisable stressed vowel, then there is no spreading, even if the intervening vowel is raisable:

  áng**o**l-o **á**ng**o**l-i 'angel (m sg/pl)' *á**u**l-i

  pé**r**s**e**g-o pé**r**s**e**g-i 'peach (m sg/pl)' *pé**r**s**i**g-i

/ pérseg-i /	*HIGHLAX	LICENSE(hi,stress)	DON'TSHARE
→ pérseg-i		*	
pérsig-i		*	*!
pírsig-i	*!		**

This shows that partial spreading is not a markedness improvement. If the feature [+high] simply wanted to be aligned to the left, for example, we would expect *[pérsig-i]. Rather, spreading improves markedness only if it reaches all the way to the stressed vowel, as LICENSE(high, stress) requires.

Markedness constraints for HS analysis

***HIGHLAX.** This one was simple. I didn't want to get involved with risky non-ASCII symbols, even though the manual says (p. 15) that I could use HTML codes. Instead, in my inputs and outputs I just used capital "I, U" to represent the high, lax vowels, including the one derived from raising /a/ (written as "I", even though it ought to be central). So, the markedness constraint is just:

[constraint]
[long name] *HighLax
[active] yes
[type] markedness
[definition] [IU]

LICENCE(high, stress). This was tougher. I couldn't think of an easy way to represent autosegmental structure, so instead I penalized a sequence of *stressed-non-high...high*. This should work if every candidate has exactly one stressed vowel, and if I assume that a sequence of high vowels shares a single [high] specification, and if the unstressed high vowel that needs to get its [+high] licensed always comes *after* the stressed vowel rather than before.

Because I'm already using capital letters for lax vowels, I had to introduce ` as a stress mark (precedes each stressed vowel).

I had to use the long-distance markedness-constraint format, with 5 tab-separated arguments L, M, C, Q, R (see p. 70 of manual):

[constraint]
[long name] License(high,stress)
[active] yes
[type] markedness
[definition] `[eEoOa] [^iuIU] [iuIU] .

[comment] formant of definition is L M C Q R
[comment] L, the left-hand context, is a stressed, non-high vowel: `[eEoOa]
[comment] M, what may be skipped, is anything other than a high vowel: [^iuIU]
[comment] (this crucially assumes only one stress per word)
[comment] C, what counts in determining violations, is a high vowel: [iuIU]
[comment] Q, what may be skipped when looking for R, is anything: .
[comment] R, the right-hand context, is null (but there is still a tab for it): (blank)

Operations for HS analysis

The most important operation is **spreading [+high] one syllable** (and only one!) to the left. This was fairly easy to define—the only tricky part was to be able skip an intervening consonant. For this, I referred to section 3.5 of the manual. Unfortunately, it seems like I have to give a separate definition row for each target vowel—that is, I can't just use [aeEoO] at the beginning, and [IiIuU] at the end, as in the Linux *tr* command. (The capital S is there because I'm using it in my representations for IPA ʃ).

[operation]
[long name] SpreadHigh
[active] yes
[definition] a_[bcdfghjklmnpqrstvwxyzS]*_[iuIU] I 2 3
[definition] e_[bcdfghjklmnpqrstvwxyzS]*_[iuIU] i 2 3
[definition] E_[bcdfghjklmnpqrstvwxyzS]*_[iuIU] I 2 3
[definition] o_[bcdfghjklmnpqrstvwxyzS]*_[iuIU] u 2 3
[definition] O_[bcdfghjklmnpqrstvwxyzS]*_[iuIU] U 2 3
[violated faith] DontShare

[comment] Structural description is.
[comment] (1) non-high vowel, (2) zero or more consonants, (3) high vowel.
[comment] Structural change is.
[comment] change (1) to the corresponding high vowel, and leave 2 and 3 alone.

Although spreading [+high] also can violate IDENT(high), I didn't worry about it, since in all my examples, DON'TSHARE and IDENT(high) would have exactly the same number of violations for every candidate. It's also conceptually weird that I'm defining DON'TSHARE as a faithfulness constraint—a better name might have been DON'TADDANASSOCIATIONLINE.

The other change I allowed was **tensing** a lax, non-low vowel:

[operation]
[long name] Tensify
[active] yes
[definition] E e
[definition] O o
[definition] I i
[definition] U u
[violated faith] Ident(tense)

I could have an operation to lower the suffix vowel if I'd wanted to further expand the candidate set.

Tableaux for HS analysis

I used the same inputs as shown in the parallel OT analysis above, but with the ASCII-friendly notation mentioned above:

[typology]
[begin tableaux]
m`ovi 0 1
v`EtSi 0 1
`ordeni 0 1
lav`oravi 0 1
p`Ersegi 0 1
[end of tableaux]

Results and discussion, including, is Venetan part of the HS typology?

There were four resulting languages predicted.

Inputs	m`ovi	v`EtSi	`ordeni	lav`oravi	p`Ersegi	*my comments on each language*
1	m`ovi	v`EtSi	`ordeni	lav`oravi	p`Ersegi	No changes at all: DON'TSHARE is top ranked
2	m`uvi	v`EtSi	`ordeni	lav`oravi	p`Ersegi	Raising, but only of a tense vowel (IDENT(tense) and *HIGHLAX are top ranked), and only of an *adjacent* stressed vowel (see discussion below)
3	m`uvi	v`ItSi	`ordeni	lav`oravi	p`Ersegi	Raising of any adjacent, stressed vowel, even if the result is lax: IDENT(tense) is ranked high, but not *HIGHLAX
4	m`uvi	v`itSi	`ordeni	lav`oravi	p`Ersegi	Raising and tensing of any adjacent, stressed vowel: *HIGHLAX outranks IDENT(tense)

Grammar 1: *HighLax, DontShare, Ident(tense) >> License(high,stress)
Grammar 2: *HighLax, Ident(tense) >> License(high,stress) >> DontShare
Grammar 3: License(high,stress), Ident(tense) >> *HighLax, DontShare
Grammar 4: License(high,stress) >> *HighLax, DontShare >> Ident(tense)

If there were a row for Veneto, it would look like this (bolding the outputs that are different from their inputs):

m`uvi	v`EtSi	**`urdini**	lav`oravi	p`Ersegi

To see why Venetan was not part of the typology, I chose the language that was closest to it, Language 2, and inspected the full derivation for /órden-i/ to see why no raising occured, even though raising does occur in /móv-i/ → [múv-i]. It had only one derivational step—that is, the output is identical to the input on the first iteration:

Input: `ordeni	*HighLax	Ident(tense)	License(high,stress)	DontShare
☞ `ordeni	0	0	-1	0
`ordini	0	0	-1	-1

In order to get from /órden-i/ to [úrdin-i], we would have to first spread [+high] by one vowel, to [órdin-i] (and then spread all the way to [úrdin-i] in the final step). But, as we see in the tableau above from OT-Help, spreading just once is harmonically bounded: it does not solve the licensing problem, and it incurs a faithfulness violation (DON'TSHARE).

Thus, as Walker observed, Harmonic Serialism predicts myopia in this type of spreading. If partial spreading is not in fact a markedness improvement, then long-distance spreading can't occur in HS, because the intermediate step of partial spreading will never be taken.