Second half of Class 6, I hope: Rule+constraint theories?

Overview: We'll try to make the framework for rule/constraint interaction more explicit (and find problems in so doing).

1. Reminder of where we left off

- People liked constraints, because
  - They allow rules within a language that do related things (like eliminate or avoid creating CCC) to share something formally (*CCC)
  - They gave clearer theoretical status to the idea of "markedness"
    - Everyone knew languages don't "like" CCC sequences (they are "marked"), but this was not directly encoded in grammars until constraints like *CCC came along.

Review of how rule application would work

2. "Normal" rule application, no constraints

- apply V → Ø / VC__CV to /bladupi/

program	*contents of* `current_form`
`current_form <- bladrupi`	`bladupi`
`current_form <- deletion_rule(current_form)`	`bladpi`
`current_form <- next_rule(current_form)` *etc., till all rules used*	`blatpi` *or whatever*
`return(current_form)`	

3. Constraints as rule blockers

- apply V → Ø / C__C , unless result would violate *CCC
  - … to /bladupi/

program	current_form	candidate_forms
`current_form <- bladupi`	bladupi	
`candidate_forms <- deletion_rule(current_form)`	bladupi	<bldupi, bladpi, bldpi>

`for i in length(candidate_forms)`
`{`
`    if (no_CCC(candidate_forms[i]) == TRUE)`
`    {`
`        current_form <- candidate_forms[i]`
`        exit loop`
`    }`
`}`
	i=1 : bladupi
	i=2 : bladpi
	(then exit)
		<bldupi, bladpi, bldpi>

apply more rules	blatpi	
`return(current_form)`		

Worry: what if there's an equally viable candidate form later in the list? What determines the order of the candidate list?

4. Constraints as rule triggers

- Ø → i , only if needed to eliminate *CCC violation
    - … to /katspa/

program	current_form	candidate_forms
current_form <- katspa	katspa	

if (no_CCC(current_form) == FALSE)
{
**   candidate_forms <-**
insertion_rule(current_form)
}
	katspa	<ikatspa, kiatspa,
		kaitspa, katispa,
		katsipa, katspia,
		katspai>

for i in length(candidate_forms)
{
**   if (no_CCC(candidate_forms[i]) == TRUE)**
**   {**
**   current_form <- candidate_forms[i]**
**   exit loop**
**   }**
}
	i=1 : katspa
	i=2 : katspa
	i=3 : kaitspa
	i=4 : katispa
	(then exit)
		<ikatspa, kiatspa,
		kaitspa, katispa,
		katsipa, katspia,
		katspai>

apply more rules	ketispe	
return(current_form)	ketispe	

Same worries: what if there's an equally viable candidate form later in the list? What determines the order of the candidate list?

5. Explicit proposals for implementing blocking and triggering?

- There weren't a lot.
- Sommerstein (1974) had a proposal for implementing triggering, which boiled down to…
  1. Check whether applying the rule would *eliminate, reduce*, or *alleviate* violations of at least one of the constraints that are listed as "motivating" that rule
  2. If so, apply it. If not, don't.

- What do those terms mean, for Sommerstein?
    - *eliminate*: ketspe → ketsipe
    - *reduce*: ketspelkno → ketsipelkno (suppose the rule was Ø → i / [–son] __ [–son])
        - ? another example to try: rule is Ø → i, constraint is *CC, form is /arbsto/

- *alleviate* is trickier, and I'm not sure there are really good cases, but here goes…
  1. If a form violates a constraint, find the smallest (fewest features) change needed to bring it into conformity
     - ? /ɑby/ violates $*\begin{bmatrix}\alpha\text{back}\\ \beta\text{round}\end{bmatrix} C_0 \begin{bmatrix}-\alpha\text{back}\\ -\beta\text{round}\end{bmatrix}$. Write the structural change needed to fix it, and count up how many features are in it:

       cost of fully repairing /ɑby/ =

  2. Now try applying the rule to the form, and find the smallest change need to bring the result into conformity
     - ? apply V → ||–round|| to /ɑby/
     - ? Write the structural change needed to fix the result, and count its features:

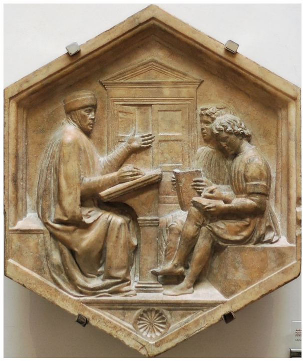
       cost of fully repairing result of applying rule to /ɑby/ =

  3. If the cost has gone down, that counts as alleviating the violation

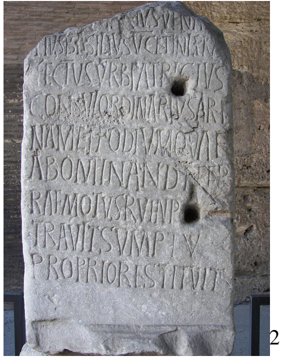
6. In case you're curious, here's the kind of case Sommerstein had in mind

Latin

- Indo-European language formerly spoken in the area around Rome in what's now Italy, and later throughout the Roman Empire
  - around 700 BCE to 700 CE
- Continued for centuries to be used for religious and scientific purposes in Europe
- Still the official language of Vatican City's government
- Source of the Latin alphabet, now used by English and many other languages
- Source of thousands of loans in English, directly and via French


[1]

Priscian, Latin grammarian


[2]

Inscription at the Colosseum

[1] en.wikipedia.org/wiki/Priscian#/media/File:Priscianus_della_Robbia_OPA_Florence.jpg

[2] en.wikipedia.org/wiki/Latin#/media/File:Rome_Colosseum_inscription_2.jpg

genitive sg.	*nominative sg.*	*UR*	
lakt-is	lak	/lakt/	'milk'
kord-is	kor	/kord/	'heart'

If we have these constraints, which are "surface-true" in Latin…

- *no final voiced in cluster* $* [+\text{consonantal}]\begin{bmatrix}+\text{consonantal}\\ +\text{voice}\end{bmatrix}\#$ (p. 82)
- *final obst. restrictions* if $\underset{1}{\begin{bmatrix}-\text{sonorant}\\ <-\text{continuant}>\end{bmatrix}}\underset{2}{[-\text{sonorant}]}\#$ then 2 is $\begin{bmatrix}+\text{coronal}\\ <+\text{continuant}>\end{bmatrix}$ (p. 82)
  - "If a word ends in two obstruents, the second one has to be coronal"
  - "…and if the first one is a stop, the second has to be not just any coronal but [s] specifically"
  - i.e., [st], [ps], [ks] are OK

… then we can have a very simple rule: C → Ø / __ #
(instead of packing all that information into the rule(s))

- A derivation might look like this (we'll fill it in):

	/lakt/	/kord/	/re:ks/
violates no final voiced in cluster*?*	no	yes	no
violates final obstruent cluster restrictions*?*	yes	no	no
if any 'yes', tentatively apply deletion			NA
is the violation alleviated/eliminated?			NA
if so, accept the change (else don't)			NA

7. Multiple available repairs

- Imagine a Roman, Caecilius, who for some reason ends up with this additional rule:
  [ ] → [–voice]

? How does our derivation change (assuming Caecilius sounds the same as other Romans)? Do we need to add more information to his grammar?

- Imagine Caecilius's neighbor, Metella, who for some reason has this rule (plus the normal Latin rules):

  [ ] → [+continuant]

? How does our derivation change (again, assuming Metella sounds like everyone else)? Do we need to add more information to her grammar?

8. Blocking vs. triggering: Myers's (1991) persistent rules

- Zulu: Bantu language (which makes it part of Niger-Congo family)
- From South Africa, about 12 million speakers
- An official language of South Africa, one of the most widely spoken and understood languages there
- Some English words that are loans from Zulu: *impala, mamba* [could be from Swahili]

Nkosazana Dlamini-Zuma ("NDZ")
anti-apartheid activist, politician

Nokutela Dube
educator, publisher, political organizer, co-founder of first Zulu newspaper

[3]

one of Lucky Dube's Zulu-language albums

Benedict Vilakazi
poet, novelist

[3] from discogs

- Zulu has prenasalized affricates (^{n}t͡ʃ, ^{n}d͡ɮ, …) but no prenasalized fricatives (*^{n}ʃ, * ^{n}ɮ). We might propose a constraint:[4]

$$*\begin{bmatrix}\text{+continuant}\\ \text{+nasal}\end{bmatrix}$$

- Here is a prefix that creates prenasalized consonants (p. 329):

singular	*plural*	
uː-ba^{m}bo	izi-^{m}ba^{m}bo	‘rib’
uː-p^{h}ap^{h}e	izi-^{m}pap^{h}e	‘feather’
ama-t^{h}at^{h}u	ezi-^{n}tat^{h}u	‘three’
uː-k^{h}uni	izi-$^{\text{ŋ}}$kuni	‘firewood’

? Assume the underlying form of the prefix is /izin/. Formulate a rule or rules to cause prenasalization.

- Here’s what happens when the prefix attaches to a fricative-initial stem:

singular	*plural*	
eli-ʃa	e-^{n}t͡ʃa	‘new’
uː-fudu	izi-^{m}p͡fudu	‘tortoise’
uː-sizi	izi-^{n}t͡sizi	‘sorrow’
uː-zwa	izi-^{n}d͡zwa	‘abyss’
uː-zime	izi-^{n}d͡zime	‘walking staff’
uː-ɮubu	izi-^{n}d͡ɮubu	‘groundnut’
uː-ʃikisi	izi-^{n}t͡ʃikisi	‘quarrelsome person’

? What would happen if prenasalization were subject to blocking by the constraint above?

[4] Myers actually uses something called autosegmental representations

- Myers proposes instead a “**persistent rule**”—it tries to apply at every point in the derivation, so that any time its structural description is created, it immediately gets changed.

$$\begin{bmatrix}\text{+nasal}\\ \text{+continuant}\end{bmatrix} \rightarrow \begin{bmatrix}\text{+delayed release}\\ \text{–continuant}\end{bmatrix} \qquad \text{i.e., nasal fricative} \rightarrow \text{affricate}$$

? Let’s spell out what the derivation would look like.

? Can we recast this as a simpler rule that is triggered by the constraint?

9. Interim summary

- We’ve tried to make a rules+constraints theory work, really spelling out the details.
- You should now feel uncomfortable about ignoring conspiracies, yet also uncomfortable about exactly how constraints are supposed to work.
  - Now you know how many phonologists felt through the 1970s and 1980s!

The “conceptual crisis” (Prince & Smolensky 2004, p. 1)

- Since Kisseberth 1970, constraints were taking on a bigger and bigger role. But as we saw there were open questions…

10. Why aren’t constraints always obeyed?

- Korean avoids VV and CC through allomorph selection (narrow-ish transcription):

plain	*nominative*	
ton	ton-i	‘money’
saɾam	saɾam-i	‘person’
koŋ	koŋ-i	‘ball’
namu	namu-ga	‘tree’
pʰaɾi	pʰaɾi-ga	‘fly’
kʰo	kʰo-ga	‘nose’
ɕ*i	ɕ*i-ga	‘seed’

- And yet, CC and VV occur in the language

plain	*locative*
namu	namu-e
kʰo	kʰo-e
	plural
saɾam	saɾam-dɨl
koŋ	koŋ-dɨl

11. Can different constraints prioritize rules differently?

? Grammar: {*CC, *C#, C → Ø, Ø → i} What happens to /ubt/??

I'll assign you to small groups, one per problem: prepare brief discussion of your problem. I've given suggested examples and you can add your own.

12. Simple rules → more indeterminacy

? What happens if the grammar has a rule ∅ → i (with no context) and a constraint *CCC?

/arbso/

? What happens if a grammar has rules ∅ → i and C → ∅ and a constraint *CC?

/eldu/

13. What happens if there's more than one way to satisfy a constraint?

? Grammar: {*CC, C → Ø, Ø → i} What happens to /absko/??

14. What happens when constraints conflict?

- What if one constraint wants to trigger a rule, but another wants to block it?

? Grammar: {*VV, *ʔ$\begin{bmatrix}\text{V}\\ -\text{stress}\end{bmatrix}$, Ø →ʔ}[5] What happens to /aórta/?? /xáos/??

A question that came up in Perusall: would the order *in which the constraints are considered matter?*

15. Should a rule be allowed to look ahead in the derivation to see if applying alleviates a constraint violation? (how far?)

? Grammar: {*C#, C → [–voice], [–voice] → Ø} What happens to /tab/??

[5] based on Dutch; data from Booij 1995 via Smith 2005)

16. Relatedly, is a rule allowed to make things *worse* if a later rule will make them better?

? Grammar: {*CCC, Ø → p / m__s, $\begin{matrix}C & C & C & C\\ 1 & 2 & 3 & 4\end{matrix} \rightarrow 3$ ("if you have 4 consonants in a row, delete all but the third one")} What happens to /almso/??

17. Can a constraint prohibit a certain type of <u>change</u>, rather than a certain structure?

18. Where does this leave us?

- Tormented, I hope!
- It seems like constraints would be a good thing
- But we don't know how to make them work with rules and each other
- Now you know how it felt to be a phonologist in the 1970s and 1980s
- The response that took the field by storm: get rid of the rules altogether!

Coming up:

- Next reading is excerpts from Prince & Smolensky's 1993 manuscript introducing Optimality Theory (OT), an all-constraint theory.
- Over the next couple of classes we'll cover the fundamentals of OT.
  - Excruciating-detail style again, so even if you already know OT I hope you'll gain some new insights
- Then we'll move into explore the differing **predictions** that SPE, OT, and their variants make about phonologies.

References

Booij, Geert. 1995. *The phonology of Dutch*. Oxford: Clarendon Press.

Myers, Scott. 1991. Persistent rules. *Linguistic Inquiry* 22. 315–344.

Prince, Alan & Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Malden, Mass., and Oxford, UK: Blackwell.

Smith, Jennifer L. 2005. *Phonological Augmentation in Prominent Positions*. 1 edition. New York: Routledge.

Sommerstein, Alan. 1974. On phonotactically motivated rules. *Journal of Linguistics* 10. 71–94.