

Optimality Theory Introduction

From Rules to Constraint Based Grammars
Make Up Lecture

special thanks to Gaja Jarosz for some material

Goal

- To set you up for success when you learn about/work with OT in upper level linguistic courses in our department
 - This is a *very* basic introduction
- Familiarize you with
 - the idea behind OT
 - the parts of OT
 - including some terminology
 - the way the parts work together

What linguistic theory is after

- Characterizing what is and is not a possible human language
 - What are the legal combinations of linguistic units?
 - Impossible vs. Possible mappings
 - We study particular languages to get at what the universal principles might be.
 - What does that knowledge look like?

Pluralization Revisited

Spoken Forms:

- kæt+s → kæts
- dog
- klæs

Make a noun plural by adding an “s” sound.

Pluralization Revisited

Spoken Forms:

- kæt+s → kæts
- dog+s → dogz
- klæs

Make a noun plural by adding an “s” or a “z” sound.

Pluralization Revisited

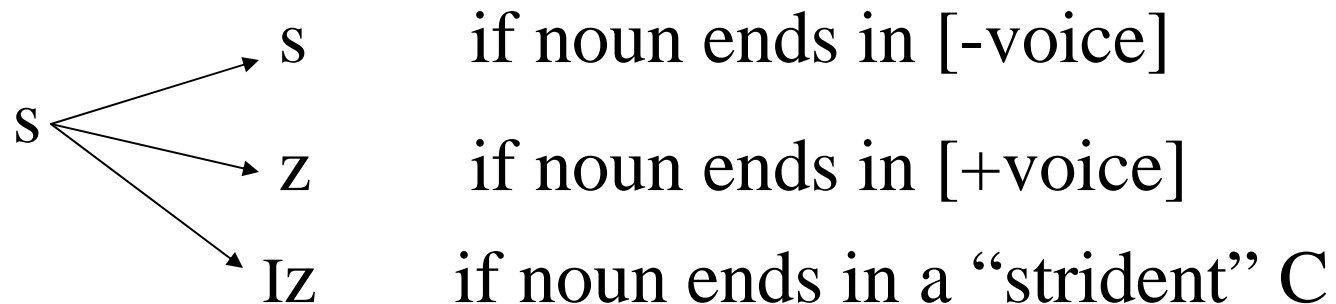
Spoken Forms:

- kæt+s → kæts
- dog+s → dogz
- klæs+s → klæsɪz

Make a noun plural by adding an “s” or a “z” or an “iz” sound.

A Single Underlying Form

- At spoken level: variations on plural marker
 - Systematic variation
- At mental representation level: single plural marker



Phonological Re-write Rules

Voicing assimilation: $s \rightarrow z / [+voice] \underline{\quad}$

Epenthesis: $\emptyset \rightarrow [I] / C \underline{\quad} C$

Phonological Re-write Rules

dog+s → dogs
→ dogz Voicing assimilation: s → z

klæs+s → klæss
→ klæsɪs Epenthesis: ∅ → [ɪ] / C_C
→ klæsɪz Voicing assimilation: s → z

Generalizations About the Input

- Other examples of allomorphy from class
indefinite article: a/an
past tense: -ed/-d/-t
- One underlying form
- Surface forms result from rule application

Generalizations in the Output

- However, application of some rewrite rules seem subject to certain conditions on the output.
 - e.g. Kisseberth (1970), Paradis (1988), Myers (1991)
- Rules + output constraints
 - something like:
 - $V \rightarrow \emptyset / C_C$
 - Condition: block if result violates constraint *CCC
 - “Do X, Unless Y”
 - “Do X, Only If Z”

Generalizations in the Output

- Appear to be generalizations at the surface level
- Early Child Phonology (some utterances of Timmy)

[pæ]	'book' [11 months]
[hək ^h a]	'key' [11 months]
[bæ:]	'bird' [15 months]
[ka]	'cup' [15 months]
[ʔəma]	'moon' [15 months]
[pæ]	'baby' [16 months]
[pæ]	'block' [16 months]
[pæ]	'boat' [16 months]
[kak ^h i]	'cookie' [16 months]
[nʌmæ]	'Simon' [16 months]

Note: child language is fraught with variation – it's not in general possible to predict the exact output for each word, but it is possible to narrow the possibilities to a small set. Focus here on what processes are observed rather than on predicting exactly which process applies in a particular situation.

Generalizations in the Output

- Appear to be generalizations at the surface level
- Early Child Phonology (some utterances of Timmy)

[pæ]	'book' [11 months]	coda drop
[hək ^h a]	'key' [11 months]	insert syllable, shorten diphthong
[bæ:]	'bird' [15 months]	coda drop
[ka]	'cup' [15 months]	coda drop
[ʔəma]	'moon' [15 months]	epenthesis
[pæ]	'baby' [16 months]	unstressed syllable drop
[pæ]	'block' [16 months]	C-cluster simplification, coda drop
[pæ]	'boat' [16 months]	coda drop
[kak ^h i]	'cookie' [16 months]	
[nʌmæ]	'Simon' [16 months]	coda drop

Note: child language is fraught with variation – it's not in general possible to predict the exact output for each word, but it is possible to narrow the possibilities to a small set. Focus here on what processes are observed rather than on predicting exactly which process applies in a particular situation.

Different Means to the Same End

- Pressure towards CV syllables: coda dropping, consonant-cluster simplification, epenthesis
 - other observed processes (e.g. devoicing) preserve CV syllable structure
- The conspiracy puzzle:
Why should all the rewrite rules conspire together to achieve the same output?

Optimality Theory

Prince & Smolensky (1993)

- Competition among ranked constraints
- A surface form is the resolution of conflicting constraints
 - not the result of rewrite rules
- Predicts “Do X, Unless Y” and “Do X, Only If Z” type patterns
- Focus on how forms surface
 - vs. how forms change

an analogy

- We deal with conflicting constraints all the time:
 - John, Mary, and Jane are meeting up for lunch.
 - John has Celiac's (no gluten)
 - Mary is dabbling with becoming vegetarian
 - Jane broke up with one of the waiters at One World
 - Mary doesn't want to drive

/input/	gluten-free options	avoid seeing ex	walking distance	tasty vegetarian options
Sam's Bagels in Charles Village	*	ok	ok	ok
Ruby Tuesday's in Charles Village	ok	ok	ok	*
Sushi Hana in Towson	ok	ok	*	ok
One World in Roland Park	ok	*	ok	ok

Two Families of Constraints

MARKEDNESS Constraints

- Make your output as unmarked as possible
 - minimizing cognitive/articulatory effort
 - maximizing discriminability
- How do we know what patterns are unmarked?
 - evidence from language typology
 - evidence from developmental trajectory


FAITHFULNESS Constraints


- Make your output as faithful to the input as possible
- necessary to prevent us from just saying “ba”

- Constraints are posited to be universal; languages are organized differently b/c they differ wrt constraint ranking.

- Prediction 1: mappings that violate faithfulness AND result in a more marked pattern are impossible
- Prediction 2: We should see real-life languages that correspond to different rankings.

Reading Constraint Tableaux

/input/	constraint 1	constraint 2	constraint 3
output candidate 1	*!		*
 output candidate 2		*	***
output candidate 3		**!	*

- The input is listed in slashes '/' in the top leftmost cell of the tableau
- The pointing finger '' identifies the winning output candidate.
- Constraints are ranked left to right (highest to lowest)
- The constraint ranking above can be written: constraint 1 » constraint 2 » constraint 3
- A '!' indicates the fatal violation for the candidate (so the winning candidate should not have a '!' in its violation profile)
- How many possible rankings of the three constraints are there?
- Which output candidate wins under each of the rankings?


An OT Analysis

Pluralization of “dog” revisited:

Markedness Constraints:
*[+voice][-voice]

Input: dog+s

Outputs: dogs, dogz, . . .

/dogs/	*[+voice][-voice]	FAITH
dogs	*!	
 dogz		*
<i>(more . . .)</i>

An OT Analysis

Pluralization of “cat” revisited:

Markedness Constraints:
*[+voice][-voice]

Input: kæt+s

Outputs: cats, catz, . . .

/kæts/	*[+voice][-voice]	FAITH
☞ kæt+s		
kætz		*!
(more . . .)

Consider the opposite ranking

Pluralization of “dog” when
FAITH dominates $*[+voice][-voice]$

Under this ranking, it's more important to be faithful, so the faithful candidate wins. This isn't the ranking in English.

Input: dog+s

Outputs: dogs, dogz, . . .

/dogs/	FAITH	$*[+voice][-voice]$
☞ dogs		*
dogz	*!	
(more . . .)

Consider the opposite ranking

Pluralization of “cat” when

FAITH dominates $*[+voice][-voice]$

kæts → *kætz* is an impossible mapping, no matter what the ranking, because it is both less faithful AND more marked.

Input: *kæt+s*

Outputs: *cats*, *catz*, . . .

/kæts/	FAITH	$*[+voice][-voice]$
☞ <i>kæts</i>		
<i>kætz</i>	*!	
<i>(more . . .)</i>

- Wait, what about “classes”?

- Originally, the plan was to work it out in the homework. Since I didn't spend as much time as planned on the last slides, I have replaced that question. If you like, you can try to work it out yourself using:

- *[+voice][-voice] – don't have a voiced segment followed by an unvoiced segment

- *[+strident][+strident] – don't have two stridents in a row

- *Epenthesis – don't add segments to the output that weren't in the input

- Faith-voice – don't change the voicing in the output to be different from that in the input

- Remember, though, this is a toy example intended to illustrate how constraint interaction works. The real story for English plurals will be more refined.

Summary

- OT posits that . . .
 - language involves constraints, not rules
 - the constraints are violable
 - often the constraints are in conflict
 - each language ranks the constraints in importance
 - a surface form is optimal if it incurs the least serious violations