Today's topic:

- **Replace *rules* with *goals*: Intro to Optimality Theory**

Background preparation:

- *Handout - "Phonology with 'goals': OT"*

0. Today's objectives

After today's class, you should be able to:

- Determine priorities between goals by identifying cases where goals conflict
- Informally describe goals/constraints and priorities seen in phonological data sets
- Use OT concepts as part of our phonological model
  - Formalize "goals" as constraints
  - Use a constraint tableau to show an input, competing outputs, and constraint violations

0. Course registration!

- A list of LING courses offered for Fall 2026
- Posters / more information about selected courses

1. Phonology with “goals” instead of rules

- Last time, we discovered some questions and problems with a rule-based approach to building syllable structure
- An alternative model of the mental grammar has **no phonological rules**
- Instead, we can propose:
  - A universal set of **goals** that all languages share
  - A way for each language to **prioritize** conflicting goals (this allows languages to be different)

1. Phonology with “goals” instead of rules

- Under this approach, what we need to propose in analyzing a language’s phonology is not a set of rules, but a **prioritization of the universal goals**

- The goal-based phonological model we will pursue is known as **Optimality Theory (OT)**

2. How to determine priorities

- Imagine: Some pop-psychology influencer proposes that five basic drives explain all human behavior

  LOVE MONEY FAME

  EXCITEMENT POWER

  - Different people's behavior is said to be explained by how they prioritize the basic drives

- *Disclaimer:* This example is very, very fake, but we're using it to illustrate how we're going to **construct arguments** about goals in phonology

2. How to determine priorities

- How could we figure out what a given individual prioritizes?
  - Example: Does Pat prioritize LOVE or MONEY?

2. How to determine priorities

- Let's ask Pat to choose between these two job offers

	LOVE	MONEY
Job #1: Same city as SO | $45,000		
Job #2: Far from SO | $125,000		

  - What can we learn about Pat's priorities for LOVE vs. MONEY?

- Notation: **Dashed** line between goals means no claim about priority is being made

2. How to determine priorities

- Let's ask Pat to choose between these two job offers

	LOVE	MONEY
Job #1: Same city as SO | $45,000	*better*	*worse*
Job #2: Far from SO | $125,000	*worse*	*better*

  - What can we learn about Pat's priorities?
    - If Pat picks Job #1: LOVE » MONEY
    - If Pat picks Job #2: MONEY » LOVE

- Notation: **»** means "has priority over"

2. How to determine priorities

- Let's ask Pat to choose between these two job offers

	LOVE	MONEY
Job #3: Same city as SO | $125,000		
Job #4: Far from SO | $45,000		

  - What can we learn about Pat's priorities for LOVE vs. MONEY?

2. How to determine priorities

- Let's ask Pat to choose between these two job offers

	LOVE	MONEY
Job #3: Same city as SO | $125,000	*better*	*better*
Job #4: Far from SO | $45,000	*worse*	*worse*

  - What can we learn about Pat's priorities?
    - Job #3 is better for both LOVE and MONEY, so we know Pat will pick Job #3
    - BUT: We **don't know** anything about **priorities**, because there is **no conflict**

2. How to determine priorities

- Let's ask Pat to choose between these two job offers

	LOVE	MONEY
Job #5: Same city as SO | $45,000 | Chapel Hill, NC		
Job #6: Far from SO | $125,000 | Paris, France		

  - What can we learn about Pat's LOVE vs. MONEY...
    - if Pat picks Job #5?
    - if Pat picks Job #6?

2. How to determine priorities

- Let's ask Pat to choose between these two job offers

	LOVE	MONEY	EXCITEMENT
Job #5: Same city as SO | $45,000 | Chapel Hill, NC	*better*	*worse*	*worse*
Job #6: Far from SO | $125,000 | Paris, France	*worse*	*better*	*better*

  - What can we learn about Pat's LOVE vs. MONEY...
    - if Pat picks Job #5? | LOVE » MONEY
    - if Pat picks Job #6? | we can't tell!

2. How to determine priorities

- What we've noticed so far:

  If we want to test the **relative importance** of LOVE vs. MONEY...

  - The two scenarios need set up a **conflict**, so that one wins by LOVE and the other wins by MONEY
  - We need to be sure there isn't some **third factor** that we aren't thinking about that's really the reason for the choice being made (EXCITEMENT)

2. How to determine priorities

- Here's a place where our metaphor breaks down…

	LOVE	MONEY
Job #7: Same city as SO | $45,000		
Job #8: Far from SO | $45,001		

  - What does the theory predict that Pat will pick **if** MONEY » LOVE?
  - Is this plausible human behavior?
  - This is, in fact, what Optimality Theory predicts will happen in a phonological goals scenario!

3. Trying this out in actual phonology

- Data sets: English vs. Cairene Arabic
  - When we analyzed English before, what syllable structure did we assign to this form? (Why?)

    /æklejm/ → [əkʰlejm] 'acclaim'
  - When we analyzed Cairene before, what syllable structure did we assign to this form? (Why?)

    /Ragleːn/ → [RAGleːn] 'two men'

3. Trying this out in actual phonology

- Data sets: English vs. Cairene Arabic

  English /æklejm/ → [ə.kʰlejm] ‘acclaim’ (V.CCV)

  Cairene /Ragleːn/ → [RAG.leːn] ‘two men’ (VC.CV)

Discussion

- What **goal** does English seem to be prioritizing? What about Cairene?
  - Hint: Cross-linguistic patterns in σ structure?

3. Trying this out in actual phonology

- Data sets: English vs. Cairene Arabic
  - English: Goal seems to be “Don’t have a coda”
  - Cairene: Is the goal more likely to be “Have a coda” or “Don’t have an onset cluster”?

    How can we tell?

3. Trying this out in actual phonology

- Cairene: Is the goal more likely to be "Have a coda" or "Don't have an onset cluster"?

  How can we tell?

- The goals are **universal** (=present in *all* languages)
  - What would a language look like if "Have a coda" were its **top goal**?
  - What would a language look like if "Don't have an onset cluster" were its **top goal**?
  - Which of these two hypothetical language patterns is more plausible?

3. Trying this out in actual phonology

- The goals are **universal** (=present in *all* languages)
  - What would a language look like if "Have a coda" were its **top goal**?
    → All syllables would always have codas
  - What would a language look like if "Don't have an onset cluster" were its **top goal**?
    → No syllables would ever have onset clusters
  - Which of these two hypothetical language patterns is more plausible?
    (Hint: Are codas ever *required*?)

4. Formalizing goals as constraints

- Remember our discussion about how “[m] is voiced” and “[m] is [+voice]” do not mean the same thing?
  - “Voiced” is a fact about the physical world
  - [+voice] is a claim about the mental grammar
- Similarly, we need to incorporate the ideas of a phonological “goal,” and “priorities” among goals, into our **model** of the mental grammar

4. Formalizing goals as constraints

- "Goals" are formalized in OT as **constraints**
  - To propose a constraint, we need to give it a **formal definition** stating the conditions under which that constraint assigns a **violation** ('*')
  - Constraint definitions refer to the entities in our model of phonological representations, such as features, word boundaries, syllable structure, …
  - It is useful to give the constraint a convenient **name**, and provide a paraphrase of what goal it represents, but the **definition is key**

4. Formalizing goals as constraints

- "Goals" are formalized in OT as **constraints**
  - Ideally, each constraint formalizes one **simple** goal
  - Complicated patterns should come from the **interaction** of simple constraints, not from constraints that are themselves complex

4. Formalizing goals as constraints

Discussion

- How would we make a precise statement of the conditions under which each of these constraints assigns a violation?
  - NoCoda
    Assign one * for every…
  - NoOnsetCluster (hint: what is a “cluster”?)
    Assign one * for every…

4. Formalizing goals as constraints

Debriefing

- How would we make a precise statement of the conditions under which each of these constraints assigns a violation?
  - NoCoda

    Assign one * for every syllable that has a coda
  - NoOnsetCluster (hint: what is a "cluster"?)

    Assign one * for every syllable that has more than one segment in the onset

5. Constraint tableaus: UR to surface form

- How does the mental grammar **use constraints** to get from a UR to the appropriate surface form?

  Here's a big change from rule-based phonology

5. Constraint tableaus: UR to surface form

- How does the mental grammar **use constraints** to get from a UR to the appropriate surface form?

  Here's a big change from rule-based phonology

  - In OT, the grammar **does not** *turn* URs into surface forms by *changing* them step-by-step
  - Instead, the grammar…
    - takes a **UR**
    - generates a **set of potential surface forms**
    - **picks the best option** (according to the **constraints** and how they are **prioritized**)

5. Constraint tableaus: UR to surface form

- What is "a set of potential surface forms?"
  - For now, we'll start from this simplified position:
    - *English could have been like Cairene, but it isn't*
    - *Cairene could have been like English, but it isn't*

    Each language's **real** surface form is a "potential" one for the other language

Language	*Real (winning) SF*	*Another potential SF*
English	[ə.k^{h}lejm]	[ək.lejm]
Cairene	[RAG.leːn]	[RA.gleːn]

5. Constraint tableaus: UR to surface form

- How does the grammar use constraints to pick the "best" surface form for a given UR?

- For each **input** (think UR for now), the grammar creates a **constraint tableau**, which contains:
  - All the **candidate output** forms, including the **winning**, or **optimal**, output (surface form)
  - All the **constraints**
  - **Violation marks** assigned by each constraint to every candidate

5. Constraint tableaus: UR to surface form

- Here is our mini-example from English
  - Input in the top left corner
  - Constraints across the top
  - Output candidates down the side
  - The winning candidate is indicated with '→'
    (another common notation is '☞', a pointing finger)

/æklejm/	NoCoda	NoOnsetCluster
(a) [ək.lejm]		
→ (b) [ə.k^{h}lejm]		

5. Constraint tableaus: UR to surface form

- Here is our mini-example from English
  - How do these constraints assign violations here?

/æklejm/	NoCoda	NoOnsetCluster
(a) [ək.lejm]		
→ (b) [ə.k^{h}lejm]		

5. Constraint tableaus: UR to surface form

- Here is our mini-example from English
  - How do these constraints assign violations here?

/æklejm/	NoCoda	NoOnsetCluster
(a) [ə**k**.lej**m**]	**	
→ (b) [ə.**k^{h}l**ej**m**]	*	*

5. Constraint tableaus: UR to surface form

- Here is our mini-example from Cairene
  - How do these constraints assign violations here?

/Ragleːn/	NoCoda	NoOnsetCluster
→ (a) [RAG.leːn]		
(b) [RA.gleːn]		

5. Constraint tableaus: UR to surface form

- Here is our mini-example from Cairene
  - How do these constraints assign violations here?

/Ragleːn/	NoCoda	NoOnsetCluster
→ (a) [RA**G**.leː**n**]	**	
(b) [RA.**g**leː**n**]	*	*

5. Constraint tableaus: UR to surface form

- Once we have our constraint tableau with...
  - the input (UR)
  - a set of candidates for the output (surface form)

  ...then the constraints can pick a surface form,
  based on how they are prioritized
- You will practice thinking about constraint priorities in the prep questions for next time
- Next class, we will formalize priorities as **constraint ranking** in our phonological model