Objectives:

- **Find informative losers**
- **Make valid ranking arguments with comparative tableaus**

Background preparation:

- *Consider other "goals" for English VCCV*

0. Today's plan

- Quick review: Where we are with OT
- Informative losing candidates
- Comparative tableaus
- Practice with W/L notation
- Adding constraints to the analysis

1. Review: Some OT fundamentals

- In Optimality Theory (OT), we formalize
  - “phonological goals” as …
  - “priorities among goals” as …
- Universal, or language-specific?

- In principle, analyzing the phonology of a language means determining …

1. Review: Some OT fundamentals

- In Optimality Theory (OT), we formalize
  - "phonological goals" as **constraints**
  - "priorities among goals" as a **constraint ranking**
- Universal, or language-specific?
  - Constraints are **universal**
  - Constraint rankings are **language-specific**
- In principle, analyzing the phonology of a language means determining **its constraint ranking**

  … but we are simultaneously trying to figure out what constraints are in the universal constraint set

1. Review: Some OT fundamentals

- What information goes into a **constraint tableau** when we want to know how constraints are ranked?

1. Review: Some OT fundamentals

- What information goes into a **constraint tableau** when we want to know how constraints are ranked?
  - **Input** (for now, this is the same as a UR)
  - The **winning output** (the actual surface form)
  - **Competing output candidates** (possible SRs)
  - **Constraints**
  - Constraint **violations** for each candidate
- In OT, the mental grammar
  - **does not** use rules to change URs step-by-step
  - **does** use **constraints** to **choose** the best SR

1. Review: Some OT fundamentals

- What do we mean by saying that the candidates in a tableau are "**all** the **possible** SRs"?
  - For now, assume this means "any SR that **some language would plausibly pick** for this input"
  - We will come back to this question again later

2. Today's focus

- How many candidates do we need to **show** in a tableau when we are doing an OT analysis?
  - Focus on **informative losers** — losing candidates that show us something about **how constraints are ranked**
    - Remember "LOVE vs. MONEY"?
  - Informative losers can also tell us something about **what the universal constraints are**
    - Some constraint has to *make* them lose!

2. Today's focus

- Today, we will revisit Cairene Arabic and English and…
  - Use informative losers to develop **valid ranking arguments**
  - Identify additional **informative losers**
  - Use those informative losers to propose some **new constraints**

3. Ranking arguments: Comparative tableaus

- The **ranking tableau** shown here presents our analysis of Cairene Arabic from last time: with these two constraints, ranked as shown

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

- Loser *[RA.gleːn] is **plausible** & **informative**—Why?

3. Ranking arguments: Comparative tableaus

- Cairene Arabic (ranking tableau):

/Raglaːn/	NoOnsetCluster	NoCoda
→ (a) [RAG.laːn]		**
(b) [RA.glaːn]	*	*

- Loser *[RA.glaːn] is **plausible** & **informative**—Why?
  - Plausible: Some languages would choose it
  - Informative: It sets up **constraint conflict** between NoOnsClust and NoCoda

3. Ranking arguments: Comparative tableaus

- A **comparative tableau** is a way of notating a tableau to make **constraint conflict** explicit
  - This lets us identify a **valid ranking argument**
- A **valid ranking argument** identifies a constraint ranking that **must** be part of the language we are analyzing, in order for the correct candidate to win
  - Only such necessary rankings can be **proven** as part of our analysis of a language
  - Be careful not to claim more rankings than you can prove!

3. Ranking arguments: Comparative tableaus

- Cairene Arabic:

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

- Last time, we used the "worse"/"better" notation to help us identify constraint conflict
- Now we will expand on this idea:
  Compare **every** loser in a tableau to the winner

3. Ranking arguments: Comparative tableaus

- A **comparative tableau** shows "W" and "L" marks in the row for **each loser**
  - Compare the winner to each loser, one at a time
  - For **each constraint**, ask:
    - Does it think **the winner** is better? If so, add **W**
    - Does it think **this loser** is better? If so, add **L**

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

3. Ranking arguments: Comparative tableaus

- A **comparative tableau** shows "W" and "L" marks in the row for **each loser**
  - Compare the winner to each loser, one at a time
  - For each constraint, ask:
    - Does it think **the winner** is better? If so, add **W**
    - Does it think **this loser** is better? If so, add **L**

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

3. Ranking arguments: Comparative tableaus

- A **comparative tableau** shows "W" and "L" marks in the row for **each loser**
  - Compare the winner to each loser, one at a time
  - For each constraint, ask:
    - Does it think **the winner** is better? If so, add **W**
    - Does it think **this loser** is better? If so, add **L**

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

3. Ranking arguments: Comparative tableaus

- A **comparative tableau** shows "W" and "L" marks
  - If a constraint with L is ranked too high, it will pick the loser — "dangerous" for our analysis
  - **Every L constraint must be dominated by at least one W constraint** (from the same tableau row)

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

  - This confirms our NoOnsClust » NoCoda ranking

4. Practice: W/L marks and informative losers

Group discussion

- Here is the analysis we developed for **English**
  - How would we add W/L marks to this tableau?

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

4. Practice: W/L marks and informative losers

- Here is the analysis we developed for English
  - How would we add W/L marks to this tableau?

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

  - This confirms our NoCoda » NoOnsClust ranking

- In English, (a) defeats (b) because avoiding codas is more important than avoiding onset clusters

4. Practice: W/L marks and informative losers

/æklejm/	NoCoda	NoOnsetCluster
→ (a) [ə.kʰlejm]	*	*
(b) [ək̲.lejm]	** W	L

Group discussion

- Find one or more (losing) output candidates for input /æklejm/ (don't worry about aspiration) that avoid having [k̲] as a coda **in some other way** besides putting the [k̲] in an onset cluster

4. Practice: W/L marks and informative losers

- Assign W/L marks to these new informative losers

/æklejm/	NoCoda	NoOnsetCluster
→ (a) [ə.klejm]	*	*
(b) [ək.lejm]	** W	L
(c) [ə.kə.lejm]	*	
(d) [ə.lejm]	*	

4. Practice: W/L marks and informative losers

- Assign W/L marks to these new informative losers

/æklejm/	NoCoda	NoOnsetCluster
→ (a) [ə.klejm]	*	*
(b) [ək.lejm]	** W	L
(c) [ə.kə.lejm]	*	L
(d) [ə.lejm]	*	L

- Which candidate(s) will the grammar pick here?

4. Practice: W/L marks and informative losers

- Assign W/L marks to these new informative losers

/æklejm/	NoCoda	NoOnsetCluster
(→)(a) [ə.klejm]	*	*
(b) [ək.lejm]	** W	L
× (c) [ə.kə.lejm]	*	L
× (d) [ə.lejm]	*	L

- Which candidate(s) will the grammar pick here?
  - The grammar currently picks (c) and (d), *not* (a)!

5. Adding constraints to the analysis

- **What constraints** could make (c) and (d) lose?

/æklejm/	NoCoda	NoOnsetCluster
(→)(a) [ə.klejm]	*	*
(b) [ək.lejm]	** W	L
× (c) [ə.kə.lejm]	*	L
× (d) [ə.lejm]	*	L

5. Adding constraints to the analysis

- **What constraints** could make (c) and (d) lose?
  - We need a constraint **against deletion**
  - We need a constraint **against epenthesis**

- How can we define these constraints?
  - Assign one * for every …

5. Adding constraints to the analysis

- **What constraints** could make (c) and (d) lose?
  - We need a constraint **against deletion**
  - We need a constraint **against epenthesis**

NoDeletion	Assign one * for every segment in the input that is not in the output
NoEpenthesis	Assign one * for every segment in the output that is not in the input

- Are these plausible constraints?
  - Is avoiding deletion/epenthesis a plausible goal?

6. For next time

- The prep questions ask you to try something new:
  - **Given** a particular constraint ranking …
  - … Which output candidate is **predicted** to win?

/ INPUT /	ConstraintA	ConstraintB
(a) [OUTPUT #1]	***	
(b) [OUTPUT #2]	**	*