Today's topics:

- **Informative losers**
- **Valid ranking arguments**

Background preparation:

- Prep qns: English vs. Cairene Arabic VCCV

0. Today's objectives

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

- Set up a constraint tableau with input, outputs, constraints, and violation marks (review)
- Identify informative losing candidates
- Use W/L notation to make valid ranking arguments
- Add additional relevant candidates and constraints to analyze further aspects of a language's ranking

1. Warm-up: Some OT basics

- 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. Warm-up: Some OT basics

- 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. Warm-up: Some OT basics

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

1. Warm-up: Some OT basics

- 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

1. Warm-up: Some OT basics

- What is different about how the mental grammar gets from the UR to the SR in OT, compared to our previous model?

1. Warm-up: Some OT basics

- In OT, the mental grammar
  - **does not** use **rules** to **change** a UR step-by-step into its SR
  - **does** use **constraints** to **choose** the best SR for a given UR

2. More 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. More OT fundamentals

- Informally, we said that all languages share the **same phonological goals**, but these are **prioritized differently** in different languages

- We **formalize** these ideas (=incorporate them into our **model** of the mental grammar) as follows:
  - "Goals" are **constraints** with explicit definitions that refer to syllable structure, features, sonority and other elements in the mental grammar
  - "Priorities among goals" are formalized as …

2. More OT fundamentals

- We **formalize** these ideas (=incorporate them into our **model** of the mental grammar) as follows:
  - "Goals" are **constraints** with explicit definitions that refer to elements in the mental grammar
  - "Priorities among goals" are formalized as a **ranking** among the constraints
  - Example: CONSTRAINT1 » CONSTRAINT2
    - The symbol ' » ' (or '>>') means 'dominates, outranks, has higher priority than'

2. More OT fundamentals

- How does the grammar use constraints to pick the *best* SF (**optimal candidate**) for a given UR (**input**)?
- For each **input**, the grammar creates a **constraint tableau**, which contains:
  - All the **candidate output** forms
  - All the **constraints**
  - **Violation marks** assigned by each constraint
- The grammar uses the language-specific constraint **ranking** to decide which output is best
  - Start with the highest-ranked constraint first

2. More OT fundamentals

- But! Usually, as linguists, our job is to figure out what the grammar of a language is…

  How does this work in OT?

  - In OT, there are *no rules* in the mental grammar
  - Instead, our job is to figure out **how the constraints are ranked** in a given language
    - Remember "LOVE vs. MONEY"? That's our strategy
  - At the same time, we are also still refining our understanding of what the universal set of constraints actually is

3. Valid ranking arguments

- Here is our mini-example from **English**
  - How must these constraints be ranked for the grammar to choose the right syllable structure?

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

3. Valid ranking arguments

- Here is our mini-example from **English**
  - How must these constraints be ranked for the grammar to choose the right syllable structure?

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

  - NoCoda » NoOnsetCluster is the necessary ranking: NoOnsetCluster would pick the wrong candidate, so we need NoCoda to choose first

3. Valid ranking arguments

- Here is our mini-example from **Cairene**
  - How must these constraints be ranked for the grammar to choose the right syllable structure?

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

3. Valid ranking arguments

- Here is our mini-example from **Cairene**
  - How must these constraints be ranked for the grammar to choose the right syllable structure?

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

  - NoOnsetCluster » NoCoda is the necessary ranking: NoCoda would pick the wrong candidate, so we need NoOnsetCluster to choose first

3. Valid ranking arguments

- Note: To ***present*** our analysis of Cairene, we should make sure that the **constraints are shown from left to right in rank order**

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

- Remember — notation in tableaus:
  - **Dashed** line = no ranking claimed
  - **Solid** line = (left) » (right)

3. Valid ranking arguments

- Summary: The two languages have the same constraints, but in a different ranking
  - English: NoCoda » NoOnsetCluster
  - Cairene: NoOnsetCluster » NoCoda

- **Different constraint rankings** are why different languages build syllable structure differently

4. Comparative tableau format (W/L marks)

- Cairene Arabic: NoOnsetCluster » NoCoda

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

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

4. Comparative tableau format (W/L marks)

- Cairene Arabic: NoOnsetCluster » NoCoda

/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 NoOnsetCluster and NoCoda

4. Comparative tableau format (W/L marks)

- 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*!

4. Comparative tableau format (W/L marks)

- Cairene Arabic:

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

- Above, 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

4. Comparative tableau format (W/L marks)

- 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]	*	*

4. Comparative tableau format (W/L marks)

- 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	*

4. Comparative tableau format (W/L marks)

- 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

4. Comparative tableau format (W/L marks)

- 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

5. Practice: W/L marks and informative losers

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]	**	

5. 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

6. Practice: Informative losing candidates

- We also need to consider:
  - What are some of the other ways that English /æklejm/ 'acclaim' could have avoided violating NoCoda, other than by violating NoOnsetCluster?
  - *What constraints do we need in the grammar so that these other output candidates do not win?*
- This is a typical research strategy for both…
  - determining **how constraints are ranked** in a given language
  - determing **what the set of constraints** itself is

6. Practice: Informative losing candidates

- 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!

6. Practice: Informative losing candidates

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

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

6. More about informative losing candidates

- Assign W/L marks to these new informative losers

/æklejm/	NoCoda	NoOnsetCluster
→ (a) [ə.klejm]	*	*
(b) [ə<u>k</u>.lejm]	** W	L
(c) [ə.k<u>ə</u>.lejm]	*	
(d) [ə.lejm]	*	

6. More about informative losing candidates

- 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?

6. More about informative losing candidates

- 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)!

6. For next time

- **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

- We will propose additional constraints, and talk about general types of constraints in OT