

## Objectives:

- **Find informative losers**
- **Distinguish markedness and faithfulness constraints**

---

*Background preparation:*

- *Consider other “goals” for English VCCV*

# 0. Today's plan

- Quick review: Where we are with OT
- Return to slides from last time: Formalizing priorities among goals
- Informative losing candidates
- Comparative tableau notation
- Practice with W/L notation
- Markedness and faithfulness constraints (time permitting)

# 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

## 2. Formalizing priority as constraint ranking

- Section 5 of outline from last time

## 3. Today's focus

- 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



## 3. Today's focus

- How many candidates do we need to show?
  - 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!

## 3. Today's focus

- Today, we will revisit English and Cairene Arabic and...
  - Identify additional informative losers
  - Use those informative losers to propose some new constraints
  - Use those informative losers to determine how the constraints are ranked

## 4. Informative losers and comparative tableaux

- Last time, we determined that Cairene Arabic has these two constraints, ranked as shown:

/Ragle:n/	NoONSETCLUSTER	NoCODA
→ (a) [RAG.le:n]		**
(b) [RA.gle:n]	*	*

- Loser \*[RA.gle:n] is **plausible & informative**—Why?

## 4. Informative losers and comparative tableaux

- Cairene Arabic:

/Ragle:n/	NoONSETCLUSTER	NoCODA
→ (a) [RAG.le:n]		**
(b) [RA.gle:n]	*	*

- Loser \*[RA.gle:n] is **plausible & informative**—Why?
  - Plausible: Some languages would choose it
  - Informative: It sets up **constraint conflict** between NoONsCLUST and NoCODA

## 4. Informative losers and comparative tableaux

- A **comparative tableau** is a way of notating a tableau to make **constraint conflict** explicit
  - This in turn 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

## 4. Informative losers and comparative tableaux

- A **comparative tableau** shows “W” and “L” marks in the row for **each loser**
  - Consider each loser, one at a time
  - For each constraint, ask:
    - Does it prefer the **winner**? If so, add **W**
    - Does it prefer the **loser**? If so, add **L**

/Ragle:n/	NoONSETCLUSTER	NoCODA
→ (a) [RAG.le:n]		**
(b) [RA.gle:n]	*	*

## 4. Informative losers and comparative tableaux

- A **comparative tableau** shows “W” and “L” marks in the row for **each loser**
  - Consider each loser, one at a time
  - For each constraint, ask:
    - Does it prefer the **winner**? If so, add **W**
    - Does it prefer the **loser**? If so, add **L**

/Ragle:n/	NoONSETCLUSTER	NoCODA
→ (a) [RAG.le:n]		**
(b) [RA.gle:n]	* W	*

## 4. Informative losers and comparative tableaux

- A **comparative tableau** shows “W” and “L” marks in the row for **each loser**
  - Consider each loser, one at a time
  - For each constraint, ask:
    - Does it prefer the **winner**? If so, add **W**
    - Does it prefer the **loser**? If so, add **L**

/Ragle:n/	NoONSETCLUSTER	NoCODA
→ (a) [RAG.le:n]		**
(b) [RA.gle:n]	* W	* L



## 4. Informative losers and comparative tableaux

- 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)

/Ragle:n/	NoONSETCLUSTER	NoCODA
→ (a) [RAG.le:n]		**
(b) [RA.gle:n]	* W	* L

- This confirms our NoONSCLUST » NoCODA ranking

## 5. 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 <sup>h</sup> lejm ]	*	*
(b) [ək.lejm ]	**	



## 5. Practice: W/L marks and informative losers

/æklejm/	NoCODA	NoONSETCLUSTER
→ (a) [ə.k <sup>h</sup> lejm ]	*	*
(b) [ə <u>k</u> .lejm ]	**      w	L

### Group discussion

- Find one or more (losing) output candidates for input /æklejm/ (don't worry about aspiration) that avoid having [k] as a coda **in some other way** besides putting the [k] in an onset cluster
- What “goals” can make these candidates **lose**?

## 5. Practice: W/L marks and informative losers

- Assign W/L marks to these new informative losers

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

## 5. Practice: W/L marks and informative losers

- Assign W/L marks to these new informative losers

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

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

## 5. Practice: W/L marks and informative losers

- Assign W/L marks to these new informative losers

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

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

## 6. Markedness and faithfulness constraints

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

/æklejm/	NoCODA	NoONSETCLUSTER
(→)(a) [ə.klejm ]	*	*
(b) [ə <u>k</u> .lejm ]	** <b>w</b>	<b>L</b>
× (c) [ə.k <u>ə</u> .lejm ]	*	<b>L</b>
× (d) [ə.lejm ]	*	<b>L</b>

→ We will pick up the discussion here next time



## 7. For next time

- Next class, we will focus on
  - Introducing faithfulness constraints
  - Practice with finding informative losers
  - Practice with making valid ranking arguments
  - More constraints involving syllable structure