Class 16 (Apr. 10, 2002): Optimality Theory

Overview
- Conspiracies
- Intro to OT

Conspiracies (Kisseberth)
Evaluation metric of Chomsky & Halle’s *The Sound Pattern of English*: rules with fewer symbols should be more widespread among the world’s languages.

Thus, a shorter grammar (i.e., fewer text characters) for a given language is likely to be one that does a better job of representing the “linguistically significant” aspects of that language.

Using the brace notation to collapse

\[
\emptyset \rightarrow V / C \ _ \ C#
\]

\[
\emptyset \rightarrow V / C \ _ \ CC
\]

into the shorter

\[
\emptyset \rightarrow V / C \ _ \ {C,C,#}
\]

says that these rules have something significant in common.

But these rules have something in common too (they avoid CCC sequences):

\[
\emptyset \rightarrow V / C \ _ \ CC
\]

\[
C \rightarrow \emptyset / CC + \ _
\]

Kisseberth proposes using a constraint to make the rules of Yawelmani simpler:

Instead of

\[
V \rightarrow \emptyset / V C \ _ \ C \ V \\
[-long]
\]

\[
V \rightarrow \emptyset / C \ _ \ C \ _ \ C \ V \ _ \ *CCC \\
[-long]
\]

The constraint can trigger rules or block them—blocking isn’t too problematic, but triggering might be. What if a constraint triggers multiple competing rules in some cases: how do you choose which rule to apply?
Many more conspiracies were identified, giving rise to more constraints.

People liked constraints, because they gave theoretical status to markedness (beyond default values for features).

**Intro to OT**

OT sought to solve some problems with constraints:

- What happens when there’s more than one way to satisfy a constraint?
- Why aren’t constraints always obeyed?

**Rule-based architecture vs. OT architecture**

<table>
<thead>
<tr>
<th>rule-based</th>
<th>OT</th>
</tr>
</thead>
<tbody>
<tr>
<td>start with UR (from mental lexicon)</td>
<td>start with UR (from mental lexicon)</td>
</tr>
<tr>
<td>apply rules in sequence—output is fully determined at all times</td>
<td>apply all possible rules, producing a set of candidate outputs</td>
</tr>
<tr>
<td>constraints may block or trigger rules</td>
<td>constraints pick the best candidate</td>
</tr>
<tr>
<td>interaction of constraints is nonexistent or sketchy</td>
<td>constraints interact through strict domination</td>
</tr>
<tr>
<td>similarity to UR is the result of not applying too many rules and not having too many constraints</td>
<td>similarity to UR is enforced by specific constraints</td>
</tr>
<tr>
<td>end with SR (send it to the phonetic system)</td>
<td>end with SR (send it to the phonetic system)</td>
</tr>
</tbody>
</table>

**Constraints**

**Markedness** constraints are constraints on the output (SR) (they require articulatory ease, or perceptual clarity, or other “natural” drives). Example: *CCC

**Faithfulness** constraints are constraints on the relationship between the input (UR) and the output (they require similarity).

- MAX-X: don’t delete X
- DEP-X: don’t insert X
- IDENT-F: don’t change a segment’s value for the feature F

**Constraint interaction**

Languages appear to use the same set of constraints, but different languages rank them differently.

**English:** DEP-V >> *CCC
Yawelmani: *CCC >> DEP-V
Constraints interact through strict ranking: start with all the candidates, then throw out all but the ones that do the best on the top-ranked constraint. Then, of the surviving candidates, throw out all but the ones that do best on the next-ranked constraint. Continue till you run out of constraint. What’s left over is the optimal candidate(s).

This is different from mere weighting of constraints: doing very well on many lower-ranked constraints can’t save a candidate that does badly on a high-ranked constraint.

It’s like alphabetization: *azzzz* comes before *baaaa*, because letters to the left take absolute precedence over those to the right.

*The tableau*

The tableau (French, pronounced [tʰæ’bloʊ] in English, plural *tableaux* [tʰæ’bloʊ] or [tʰæ’blouz]) is a way of representing this calculation. Naturally, not all candidates can be shown; not all constraints are shown, either.

This tableau shows a *ranking argument* for *CC >> DEP-V*:

```
/at+ka/  *CC  DEP-V

a [atŋka]  *  

b [atkn]  *!
```

Important parts of the tableau:
- input
- output candidates
- pointing finger
- constraints
- asterisks
- exclamation marks
- shading

---

1 Because this set is normally infinite, Mark Ellison and Jason Eisner have developed ways to represent it as a regular expression/finite-state automaton. Constraints then operate on this compact representation of the candidate set, not by going through all the candidates one by one (which would take—literally—forever).
Example: English stray erasure
Some common markedness constraints relevant to syllabification:

- **ONSET**: syllables must have onsets
- **NOCODA**: syllables must not have codas
- ***COMPLEX**: onsets and codas should not have more than one consonant
- **SONORITYSEQUENCING**: sonority must rise in the onset and fall in the coda

<table>
<thead>
<tr>
<th>/dæmn/</th>
<th>SONORITY SEQUENCING</th>
<th>IDENT-[NAS]</th>
<th>DEP-V</th>
<th>MAX-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a [dæmn]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b [dæm]</td>
<td></td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>c [dæln]</td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d [dæm.nə]</td>
<td></td>
<td></td>
<td>*!</td>
<td></td>
</tr>
</tbody>
</table>

(dotted line means we don’t know what the ranking is)

The markedness constraint on sonority sequencing requires some change to happen, but it’s the ranking of the faithfulness constraints that determines just what that change should be.

<table>
<thead>
<tr>
<th>/dæmn+əbəl/</th>
<th>SONORITY SEQUENCING</th>
<th>ONSET</th>
<th>IDENT-[NAS]</th>
<th>DEP-V</th>
<th>MAX-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a [dæmn.ə.əbəl]</td>
<td>*!</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b [dæm.nə.əbəl]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c [dæm.ə.əbəl]</td>
<td></td>
<td>*!</td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>d [dæ.mə.əbəl]</td>
<td></td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>e [dæl.nə.əbəl]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f [dæ.mə.nə.əbəl]</td>
<td></td>
<td></td>
<td></td>
<td>*!</td>
<td></td>
</tr>
</tbody>
</table>

How do I know which candidates to include?

Some candidates are obviously unnecessary in arguing for a particular ranking:

<table>
<thead>
<tr>
<th>/at+ka/</th>
<th>*CC</th>
<th>DEP-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>a [atəka]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b [atka]</td>
<td>*!</td>
<td></td>
</tr>
<tr>
<td>c [atəkəa]</td>
<td>**!</td>
<td></td>
</tr>
</tbody>
</table>

You should try to include candidates that show every possible way to satisfy each markedness constraint in the tableau, except when you have already established that they could not win.
Here, including candidate $c$ shows that we’re missing some constraints:

<table>
<thead>
<tr>
<th>/at+ka/</th>
<th>*CC</th>
<th>DEP-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ə) $a$ [atəka]</td>
<td></td>
<td>*!</td>
</tr>
<tr>
<td>$b$ [atka]</td>
<td>*!</td>
<td></td>
</tr>
<tr>
<td>$c$ [aka]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How do I know which constraints to include?**

You need enough constraints (in the correct ranking) to ensure that the right candidate wins:

<table>
<thead>
<tr>
<th>/at+ka/</th>
<th>MAX-C</th>
<th>*CC</th>
<th>DEP-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ə) $a$ [atəka]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>$b$ [atka]</td>
<td></td>
<td>*!</td>
<td></td>
</tr>
<tr>
<td>$c$ [aka]</td>
<td>*!</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You should also include all the faithfulness constraints that are violated by any of your candidates (which should be all the faithfulness constraints that are in conflict with any of your markedness constraints).

**Preview of next time (Wed., April 17—Monday is a holiday)**

- Feet and stress
- Stress in OT

**To do for next time**

- Read Gussenhoven & Jacobs ch. 13, 14
- Read *and answer study questions about* Kisseberth article