Constraint based phonology
Coursework Notes 2
Optimality Theory

1. Basic features of optimality theory

1. Every phonological structure has an input (underlying) form and an output (surface) form. These forms may be the same or they may be different. (This much is shared with classical generative phonology.)

2. These forms are operated on by a universal set of constraints - i.e. the same set of constraints operate in all languages.

3. There are two main types of constraint:

   (a) Constraints on form of the output structure - i.e. well-formedness constraints on segments and segment organisation. These constraints are grounded in universal markedness principles (as revealed for example in typological studies).

      For example, Syllables must have vowels as nuclei or Obstruents are voiceless.

   (b) Constraints on the relationship between the input and the output aimed at the preservation of information - maintaining faithfulness of the output to the input. These constraints clearly have their basis in the need to optimise the task of language processing.

4. Constraints may conflict - i.e. with some input form, it may be impossible to satisfy simultaneously constraint A and constraint B.

   For example, given an input form like /ant/, it is not possible to have a corresponding output form which will satisfy constraint (i) A syllable must begin with a consonant at the same time as constraint (ii) Output segments must be matched with input segments.

5. Constraints can be violated because a constraint reflects a universal tendency rather than an absolute prohibition or requirement.

   For example the constraint that Obstruents are voiceless is obviously violated by a number of English segments.

Ron Brasington. January 2003
6. **Languages differ**, not because of the presence or absence of constraints from the universal set, but **because of differences in the ranking** (the ordering, the relative priorities) of the constraints in this set.

7. The **output** for a given input is the form which **optimally satisfies** (i.e. A B C D

8. The optimal **output** form is not derived from repairs/processes applied to the input. It is **selected** as the best of (perhaps) a bad job from a set of forms known as **candidates**.

9. The set of candidates for evaluation and ultimately selection is generated by a procedure known as **GEN** (=generator). For any one input, this procedure produces a theoretically infinite set of pairs of forms each comprising the input form and one potential output form. The two forms are related to one another by a unique pattern of **correspondence** between their elements (usually taken to be segments).

The objects created by GEN are thus pairings or mappings of the following general type:

```
A B C D     A B C D     A B C D     A B C D
a b c d e   a c e     a b d e     e c b a
```

These examples show that, in generating correspondence relations within candidates, anything goes, though clearly it is to be expected that the most bizarre structures will very quickly fail the selection procedure.

9. The procedure which selects the optimal candidate from the total set of candidate forms is known as **EVAL** (=evaluator) a procedure which operates as follows:

1. Make CONSTRAINTS the full list of ranked constraints
2. Make CANDIDATES the full candidate set
3. Make ACCEPTABLE CANDIDATES those members of CANDIDATES meeting first of CONSTRAINTS
4. If size ACCEPTABLE CANDIDATES = 1 then BINGO! - stop.
5. If size ACCEPTABLE CANDIDATES > 1
   make CANDIDATES the ACCEPTABLE CANDIDATES
   make CONSTRAINTS the REST OF CONSTRAINTS and go to 3
Else make CONSTRAINTS the REST OF CONSTRAINTS and go to 3

Ron Brasington. January 2003
2. An example: English plurals

To see how the model operates in practice we can usefully look at a familiar collection of English forms:

<table>
<thead>
<tr>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>kæt</td>
<td>kæts</td>
</tr>
<tr>
<td>dʒiræf</td>
<td>dʒiræfs</td>
</tr>
<tr>
<td>dɔɡ</td>
<td>dɔgz</td>
</tr>
<tr>
<td>dʌv</td>
<td>dʌvz</td>
</tr>
<tr>
<td>ræm</td>
<td>ræmz</td>
</tr>
<tr>
<td>midʒ</td>
<td>midʒiz</td>
</tr>
</tbody>
</table>

3. Potentially relevant well-formedness constraints (from the universal set)

If we assume that the underlying form of the plural suffix is /z/ - so that for example CAT\textsubscript{pl} is underlingly kæt\textsubscript{z} and FISH\textsubscript{pl} is fiʃz, then we might expect to find the following (n.b. universal) constraints involved in selecting the appropriate candidate:

- Obstruents in clusters must agree in voice (ObsVoi)
- Clusters of sibilants are not permitted (*Sib-Sib)

4. Selecting the right cats

Notice, however, that if GEN provides candidates freely, then the set of candidates offering themselves as the ouput form of kæt\textsubscript{z} will include not only the correct form kæts but also kætz, kætz iz kæz and haiz - and for that matter pigz. And unfortunately for the selection procedure, all of these but kætz satisfy both of the constraints proposed above. Clearly other demands must be met by the successful candidate. (Notice that – following common practice – we are identifying candidates using the surface form only and ignoring for simplicity the correspondence relations with the input form.)

5. Faithfulness

The problem with kæz as an ouput form for kætz lies fairly obviously in the lack of correspondence in the segmental make-up of input and output stems. Input/Output correspondences are handled by faithfulness constraints. For a missing output we need:
• **Max I/O.** Every segment in the input must have a corresponding segment in the output – i.e. there must be no loss (deletion) of structure.

With **kætiz** there is also a lack of correspondence between the input and output, though now due to the gratuitous introduction of a segment. To deal with a 'missing' input we need:

• **Dep.** Every output segment must have a corresponding segment in the input – i.e. no insertion. But notice that **pigz** has four segments each arguably corresponding to one of the four segments of **kætz**. We need also to guarantee the identity of corresponding segments:

• **Ident I/O.** Corresponding segments in input and output forms must be **identical** in feature composition.

6. **Tableaux**

The language particular **ranking** of constraints and the way in which optimal candidates are selected is usually shown using **tableaux**. Below are abbreviated tableaux for **fiʃz** and **kætz**.

Constraints are ranked left to right. The input form appears in the top left cell, candidates are listed below. The notational conventions used are: winning candidate (**»**), constraint violation (**•**), fatal constraint violation (**!**). Cells are shaded when their status is no longer relevant.

When the constraint ranking is not significant, dotted rather than solid lines divide columns.

<table>
<thead>
<tr>
<th>fiʃz</th>
<th>*Sib-Sib</th>
<th>Max</th>
<th>Dep</th>
<th>ObsVoi</th>
<th>Ident I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>fiʃz</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>fis</td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>fiʃiz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>kætz</th>
<th>*Sib-Sib</th>
<th>Max</th>
<th>Dep</th>
<th>ObsVoi</th>
<th>Ident I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>kætz</td>
<td></td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kætiz</td>
<td></td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>kæts</td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pigz</td>
<td></td>
<td></td>
<td></td>
<td><em>!</em>*</td>
<td></td>
</tr>
<tr>
<td>kæz</td>
<td></td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hæts</td>
<td></td>
<td></td>
<td></td>
<td><em>!</em></td>
<td></td>
</tr>
</tbody>
</table>

The candidates used in the tableaux have obviously been selected from those which GEN would provide in order to exemplify the workings of the model. In reality many more would need to be eliminated (by further constraints). How would you cope with **kædz** as a possible candidate?