Finite State Automata, cont.

(1) Goal today:
- Work through another example of how we could start building a finite state machine to parse morphemes, this time in Spanish
- Using finite state machines to “undo phonology” and recognize forms in which a rule has applied

(2) Some basic facts about Spanish verbs
a. Spanish verbs fall into (roughly) three idiosyncratic conjugation classes, defined by their “theme vowel” ([a], [e], or [i]).

<table>
<thead>
<tr>
<th>Class 1: [a]</th>
<th>Class 2: [e]</th>
<th>Class 3: [i] (→ [e])</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘to speak’ hablar</td>
<td>‘to eat’ comer</td>
<td>‘to live’ vivir</td>
</tr>
<tr>
<td>hábl-o</td>
<td>cóm-o</td>
<td>vív-o</td>
</tr>
<tr>
<td>hábl-as</td>
<td>cóm-es</td>
<td>vív-emos</td>
</tr>
<tr>
<td>hábl-a</td>
<td>cóm-e</td>
<td>vív-es</td>
</tr>
<tr>
<td>hábl-an</td>
<td>cóm-en</td>
<td>vív-ls</td>
</tr>
</tbody>
</table>

- 1sg suffix is always -o
- Vowel in 2sg/3sg/3pl suffix is -a or -e (depending on conjugation class)
- Vowel in 1pl/2pl suffix is -a, -e, or -i, depending on conjugation class
- We’ll ignore stress

(3) Let’s just start with a machine that can recognize infinitives, 1sg forms, and 3sg forms.
- Goal: hear hablo, output hablar-V-1sg.Pres.Ind. (etc)
- Start by drawing a machine that can recognize hablo, habla, andhablar. At what point should we output the information about what verb is involved? When should we output the information about what the particular inflection is?
- Now add another verb, cantar. What states can be shared with hablar?
- Next, add some -er verbs: comer ‘eat’ and meter ‘put’; which states can be shared with -ar verbs, and which are unique?
- Then, add an -ir verb: vivir ‘live’. The machine is getting quite complicated now!

(4) Another type of verb; spelling alternations between <g> and <gu>
- In Spanish orthography, <g> is pronounced as [x] before front vowels, and [g] elsewhere (before back vowels and consonants)
- The phoneme [g] is written <gu> before a front vowel (e.g., seguir [segir] ‘follow’)
- Some verb suffixes begin with front vowels, and some with back vowels; that means that sometimes a <u> must be inserted in the orthography

<table>
<thead>
<tr>
<th>‘to distinguish’</th>
<th>distinguir</th>
</tr>
</thead>
<tbody>
<tr>
<td>disting-o</td>
<td>distingu-imos</td>
</tr>
<tr>
<td>distingu-es</td>
<td>distingu-is</td>
</tr>
<tr>
<td>distingu-e</td>
<td>distingu-en</td>
</tr>
</tbody>
</table>

- How might we handle the <u> in our machine?

(5) Overall structure of the machine:
- Left side: “lexicon” of verb roots, all feed into a common set of terminations
- Right side: rules to interpret suffixes
An FSA to recognize forms of *hablar* ‘speak’:

Incorporating also *cantar* ‘sing’:

Now let’s an *-er* verb, *comer* ‘eat’:

And another one, *meter* ‘put’:

Next, an *-ir* verb, *vivir* ‘live’. Things are getting complex because the vowel of *-ir* verbs is sometimes [e] and sometimes [i]. In this particular diagram, the states for *vivir* have been added at the top of the diagram:
Putting this all together with *distinguir*.

Since we’re not so concerned with the part that interprets each letter of the root (we’re really more interested in making sure the roots and affixes get interpreted in the right combinations), we can also “zoom out” and act as if the roots are interpreted in just a single transition: