(A) Tenji

Braille is a tactile writing system, based on a series of raised dots, that is widely used by the blind. It was invented in 1821 by Louis Braille to write French, but has since been adapted to many other languages. English, which uses the Roman alphabet just as French does, required very little adaptation, but languages that do not use the Roman alphabet, such as Japanese, Korean, or Chinese, are often organized in a very different manner!

To the right is a Japanese word written in the tenji ("dot characters") writing system. The large dots represent the raised bumps; the tiny dots represent empty positions.

1. The following tenji words represent atari, haiku, katana, kimono, koi, and sake. Which is which? You don’t need to know either Japanese or Braille to figure it out; you’ll find that the system is highly logical.

   a. haiku
   b. sake
   c. katana
   d. kimono
   e. koi
   f. atari

2. What are the following words?

   n a c l o
2009 Solutions

(A) Tenji

a. karate  b. anime

3. Write the following words in tenji characters:

a. samurai  b. miso
The spelling tutor computes the EDIT DISTANCE between the given word spelling and the correct spelling. We use the standard definition of operations required for converting one of the two given strings into the other, where each operation is one of the following three:

- Removal of a letter.
- Insertion of a letter (anywhere in the string).
- Replacement of a letter with (any other) letter.

The spelling tutor converts the edit distance into a comment using the following scheme:

<table>
<thead>
<tr>
<th>DIST</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no comment; correct</td>
</tr>
<tr>
<td>1</td>
<td>almost right</td>
</tr>
<tr>
<td>2</td>
<td>quite close</td>
</tr>
<tr>
<td>3</td>
<td>a bit confusing</td>
</tr>
<tr>
<td>4</td>
<td>very confusing</td>
</tr>
</tbody>
</table>

The examples given in the problem do NOT show comments for the edit distance of 5 or more, because Christopher Robin never makes so many mistakes, not even in long and delicate words.

The edit distances and related comments for the given misspellings of "typo" are as follows:
The spelling tutor computes the EDIT DISTANCE between the given word spelling and the correct spelling. We use the standard definition of the EDIT DISTANCE; that is, this distance is the minimal number of operations required for converting one of the two given strings into the other, where each operation is one of the following three:
Orthography design is the process of developing an alphabet and spelling rules for a language. A good orthography has several features:

- Given a spoken word, there’s no question of how to spell it.
- Given a written word, there’s no question of how to pronounce it.
- In the modern world, it’s increasingly important that it be reasonably easy to type!

Quechua is spoken today by millions of people in Peru, Ecuador, and Bolivia, the descendents of the citizens of the Incan Empire. Quechua speakers are rapidly joining the Information Age, and both Google and Microsoft Windows now come in Quechua!

Like in English, there are more sounds in Quechua than there are letters on a keyboard, but there are ways around that. For example, we can assign one letter to multiple sounds so long as a reader can always predict, from its position in the word or from other letters in the word, which sound is meant. So if the sound [b] only ever occurs right after [m], and [p] never occurs right after [m], we can just write “p” for both, since you’ll be able to predict from the previous letter whether “p” means [b] or [p].

This “phonemic principle” is the central principle of most orthographies, not just because it reduces letters but also because our minds categorize sounds in the same way.

Here are 18 words in Cuzco Quechua, as they are pronounced but not necessarily as they are written. [q] and [χ] represent special sounds that don’t occur in English.
2009 Solutions

(C) Orthography Design

awtu     car
kanka    roasted
karu     far
kiru     teeth
kisa     nettle
kisu     cheese
kunka    neck
kusa     great
layqa    witch
oqe       spotted
qasa      frost
qasi      free
qatoχ    merchant
qatuy    to barter
qatisaχ  I will follow
qelqax   writer
qelqay   to write
qo1qe    silver
qosa      husband
qosqo     Cuzco
saqey    to abandon
saχsa     striped
seqay    to climb
sikasika  caterpillar
sipeχ    murderer
sipiy    to kill
soχtaral  sixty cents
sunka    beard
toχra     ball of ash
uyariy    to listen
uywaχ    caretaker
waleχ    a lot
weqaw    waist

Notes:
• It is quite expected that few if any contestants are going to get all 20 points. There are going to be entirely correct and well-explained answers that don’t get quite as many points as another entirely correct and well-explained answer because the latter was more thorough. (In the first version of this rubric, we found that the minimal “correct score” was about 12, give or take.)
• Some solvers will have completely misunderstood what they were supposed to do. This is too bad, but they don’t get any points for well-meaning but bizarre answers! This is a contest, rather than a homework assignment, and for some of the puzzles the puzzle is to figure out what’s being asked.
2009 Solutions

(C) Orthography Design

• Half-points may be awarded.
• It is not necessary for a complete solution that the solver chooses <u> and <i> to be basic rather than <o> and <e>. From a phonemic point of view, the label of a sound is arbitrary – these could be Smiley Face and Labialized Smiley Face for all we care – and from an orthographic point of view, it doesn’t matter, they’re all just symbols.

I. Show that we don’t need separate letters for [q] and [χ]. (3pts)

• 1a. 1pt. for noticing that they never occur in the same environments
• 1b. 1pt. for correctly specifying what these environments are.
• 1c. 1pt. for clearly explaining why this means they can be the same letter. (This explanation doesn’t have to be long, just clear.)

II. Show that we can’t represent [a] and [i] by the same letter. (3pts)

• 2a. 1pt. for noticing that they do occur in the same environments
• 2b. 1pt. for finding a minimal pair, like “karu ~ kiru” or “qasa ~ qasi”. (If they have this but not 2a., give them the point for 1a anyway, since this subsumes that.)
• 2c. 1pt. for clearly explaining why this means they have to have different letters.

III. Show that we can’t represent [a] and [e] by the same letter. (3pts)

• 3a. 1pt. for noticing that they do occur in the same environments.
• 3b. 1pt. for noticing the pair “saqey” ~ “seqay”. (If they have this but not 3a, again give them that point anyway.)
• 3c. 1pt. for clearly explaining why this means they have to have different letters.

IV. Most modern Quechua orthographies get by with only three of the five vowels [a], [e], [i], [o], and [u]. Show how this is possible. (11pts)

n a c i o
First, they should establish which sounds can’t be merged into a single letter:

- 4a. 1pt. for finding a pair kisa ~ kisu or kanka ~ kunka.
- 4b. 1pt. for finding the pair qasa ~ qosa.
- 4c. 1pt. for finding the pair kisa ~ kusa.
- 4d. 1pt. for recognizing the relevance of the pairs in II and III to this question.

Second, three points for deducing which sounds can be merged:

- 1pt. for figuring out that either [e]~[i] and [o]~[u], or [e]~[u] and [o]~[i]. (They don’t have to get both for this point.)
- 1pt. for figuring out that both of these are possible.
- 1pt. for clearly explaining how this follows from the facts above and in parts II and III – that given the minimal pairs, these two are the only solutions that don’t cause two different words to be spelled the same.

Four points are available for:

- Up to 2 points for determining the conditioning environment for the difference: [e] and [o] when next to [q] and [χ], [i] and [u] elsewhere. (1 point each for the completeness of the description and the clarity of the explanation.)

- Up to 2 points for determining, based on the alternations qelqay ~ qelqαχ, qatuy ~ qatoχ, and sipiy ~ sipεχ, that [o]~[u] and [e]~[i] is the better or more likely of the possible solutions. (1 point for noticing the pattern and 1 point for correctly deducing the right phonemicization.)
The Guarani verb consists of:
1. prefix n(d)(a)-, if negation exists;
2. person and number of the subject: a- ‘I’, o- ‘he’, ja- ‘we’, pe- ‘you (pl.)’;
3. root;
4. -(r)i, if negation exists;
5. ending -ma for past tense or -ta for future tense.

where:
- the negative prefix should start with n (rather than nd) in case the root of the verb contains any nasal sound
- the vowel a is dropped from the negative prefix in case the personal prefix starts with a vowel.
- if a future tense is to be negated, the suffix is -mo’ai, rather than *-(r)i-ta; the negative suffix is -ri after the vowel i; -i otherwise.

Part 1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>akaruma</td>
<td>I was eating</td>
</tr>
<tr>
<td>ojupita</td>
<td>He will be waking up</td>
</tr>
<tr>
<td>ndavo’omo’ai</td>
<td>I will not be taking</td>
</tr>
<tr>
<td>napékororô</td>
<td>you are not crying</td>
</tr>
<tr>
<td>ndapyhyima</td>
<td>I wasn’t catching</td>
</tr>
</tbody>
</table>

Part 2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>you are not shooting</td>
<td>ne-pe-mbokapu-i</td>
</tr>
<tr>
<td>he is not singing</td>
<td>ndo-purahei-ri</td>
</tr>
<tr>
<td>we will be eating</td>
<td>ja-karu-ta</td>
</tr>
<tr>
<td>I will not be singing</td>
<td>nda-purahei-mo’ai</td>
</tr>
</tbody>
</table>
An extractive summarizer scores each sentence according to some criteria that are correlated with being a good summary sentence. Then it picks the top 3 sentences from each story based on the sum of its scores on the different criteria.

The first criterion measures primacy. The first sentence gets 3 points, the second 2, and the third 1, to account for the likely increased importance of the initial sentences. Then multiples of .1 are added, starting with .0 for the last sentence, to break the other criteria's ties in favor of earlier sentences.

The second criterion measures recency. The last sentence gets 3 points, the second-last 2, and the antepenultimate 1, to account for that they're likely summaryish themselves.

The third criterion counts named entities in the sentence, since they're likely to be the most important actors.

The fourth criterion counts words from the title that appear in the sentence (after reducing each word to a stem; e.g. struck = strike). These sentences are likely to pertain most immediately to the topic of the story.

The fifth criterion counts named entities introduced for the first time in this sentence. The first mention of a named entity is probably important for understanding its role.

The sixth criterion counts past-tense verbs. Current information is probably more important, and past-tense verbs are slightly less likely to give new information.
2009 Solutions

(E) Summary

Solutions

Story 1, sentence 2, criterion 3 - change to 1
Story 1, sentence 2, criterion 5 - change to 1
Story 1, sentence 2, total - change to 2.3

Story 1, sentence 5, criterion 3 - change to 0
Story 1, sentence 5, total - change to 3.0

Story 2, sentence 1, criterion 1 - change to 3.9
Story 2, sentence 1, total - change to 8.9

Story 2, sentence 2, criterion 1 - change to 2.8
Story 2, sentence 2, total - change to 4.8

Story 2, sentence 3, criterion 1 - change to 0.6 (similarly change the next 6 numbers down from this one to be 0.1 less than before)
Story 2, sentence 2, total - change to 2.6 (this affects the totals for the next 6 sentences)

Story 2, sentence 5, criterion 5 - change to 2
Story 2, sentence 5, total - change to 4.4

Story 2, sentence 8, criterion 5 - change to 1
Story 2, sentence 8, total - change to 4.1

Part 2:

1.7 0 2 3 2 4 4.7
(F) Anaphora

In the following dialogues, the computer is misunderstanding the human because it is misinterpreting certain words:

Example 1

Human: There's this restaurant on Bancroft that's supposed to be really good that I heard about from my mother. Can you help me find it?

Computer: Where did you last see your mother?

Example 2

Human: My friend is looking for information about on-line movie databases. What should I tell him?

Computer: You can tell movie databases reviews of movies you've seen.

Example 3

Human: My parents are having a hard time choosing a new digital camera. How can I help them?

Computer: I'm afraid I don't understand your question. People don't help cameras.

A. What are the words that the computer is misinterpreting?

Ex1: ___him_____ Ex2: ___it______ Ex3: ___them_______

*** 1 correct: 0 points
2 correct: 1 points
3 correct: 2 points

B) What rule is the computer using to interpret those words? Interpret pronouns as referring to the last noun phrase in the previous sentence.

*** 2 points for any answer that states a generalization that applies to the final NP in each first sentence and nothing else.
(F) Anaphora

*** 1 point for any answer that applies only to the final noun in each sentence, or to some intermediate category that doesn't fit the data.

C) Give a better rule that would make the computer interpret the words correctly in these examples.

***1 point for just about anything that either works on all three given sentences or is distinct better than the computer's rule,

   e.g.:

- Interpret pronouns as referring to the previous sentence's first noun.

- Interpret pronouns as referring to a noun in the previous sentence with the same number/gender properties.

- Interpret pronouns as referring to the previous sentence's subject.

- Check for sentences of parallel syntactic structure first, and refer to a noun (phrase) in the same place if there is one.
Languages are everywhere... even in places where you don’t expect them.

Consider the “combo rules” of P-Little’s Triple-I XTreem Hyp0th3tica7 Sk8boarding Game. In it, players press a series of buttons (left, right, down, up, circle, triangle, square, and X) to make their on-screen avatar perform skateboard tricks that illustrate pro boarder P-Little’s “Triple-I” philosophy of Insane, Ill-Advised, and Impossible According to the Laws of Physics.

Underneath, the game is using the methods of computational linguistics to turn this “little language” of button presses into tricks and combos. The game uses a simple shift-reduce parser to parse button “words” into combo “sentences”.

As each button-press comes in, the corresponding symbols are placed, in order, in a buffer:

1. ↑
2. ↑ ←
3. ↑ ← ○
4. ↑ ← ○ ×

If, at any point, the rightmost symbols in this buffer match any of the patterns on the next page, they are removed and replaced with a new symbol indicating a combo. So, since SX corresponds to an “ollie”, we replace it with the new symbol Ollie.

5. ↑ ← Ollie
6. ↑ ← Ollie ○
7. ↑ ← Ollie ○ ○
8. ↑ ← Ollie ○ ○ ×
9. ↑ ← Ollie ○ Ollie
More complex combos can then be built out of simpler combos. You see in rule (e) below that Ollie and Nollie can be joined by S to make a new combo. There are also rule schemas that can create new combos out of any kind of combo. Rule (j) below says that any combo (represented by a), whether it’s an Ollie or an Inverted-360-Kickflip, can be joined with itself by a S to make a Double combo:

10. \[\uparrow \leftarrow \text{Double-Ollie}\]

<table>
<thead>
<tr>
<th>If the right side of the input matches…</th>
<th>Replace it with…</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. [\leftarrow \uparrow \Box]</td>
<td>[\rightarrow \text{Backside-180}]</td>
</tr>
<tr>
<td>b. [\rightarrow \downarrow \O]</td>
<td>[\rightarrow \text{Frontside-180}]</td>
</tr>
<tr>
<td>c. [\Box \times]</td>
<td>[\rightarrow \text{Ollie}]</td>
</tr>
<tr>
<td>d. [\times \Box]</td>
<td>[\rightarrow \text{Nollie}]</td>
</tr>
<tr>
<td>e. Nollie [\Box] Ollie</td>
<td>[\rightarrow \text{Woolie}]</td>
</tr>
<tr>
<td>f. [\downarrow \downarrow]</td>
<td>[\rightarrow \text{Crouch}]</td>
</tr>
<tr>
<td>g. Backside-180 Frontside-180</td>
<td>[\rightarrow \text{Backside-360}]</td>
</tr>
<tr>
<td>h. Crouch Backside-360</td>
<td>[\rightarrow \text{360-Kickflip}]</td>
</tr>
<tr>
<td>i. [\downarrow \text{a} \uparrow]</td>
<td>[\rightarrow \text{Inverted-\text{a}}]</td>
</tr>
</tbody>
</table>
Complex combos can get pretty involved. Here are a few combos from the manual to give you an idea:

<table>
<thead>
<tr>
<th>Combo</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverted-Nollie:</td>
<td>4XS3 4XS3 4XS3 4XS3</td>
</tr>
<tr>
<td>Double-Inverted-Woolie:</td>
<td>4XSSSX3S4XSSSX3 4XSSSX3S4XSSSX3 4XSSSX3S4XSSSX3 4XSSSX3S4XSSSX3</td>
</tr>
<tr>
<td>Inverted-Triple-Backside-180:</td>
<td>423TS23TS23T3 423TS23TS23T3 423TS23TS23T3 423TS23TS23T3</td>
</tr>
<tr>
<td>Inverted-Backside-360:</td>
<td>423T14C3 423T14C3 423T14C3 423T14C3</td>
</tr>
<tr>
<td>Triple-360-Kickflip:</td>
<td>4423T14CS4423T14CS4423T14CS4423T14CS4423T14CS4423T14CS4423T14CS4423T14C</td>
</tr>
</tbody>
</table>
1. How would you perform an “Inverted-Atomic-Backside-360”?

\[ \downarrow \leftarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \right}
When the parser comes across the next Ollie, it then combines it with the previous Double-Ollie to make a Triple-Ollie

9. Double-Ollie
10. Double-Ollie
11. Double-Ollie
12. Double-Ollie
13. Triple-Ollie

However, there’s no way to turn a Triple-Ollie into a Quadruple-Ollie. You can never get a sequence that runs Double-Ollie S Double-Ollie, because the first half of the second Double-Ollie would have already combined with the previous symbols to create a

14. Triple-Ollie
15. Triple-Ollie
16. Triple-Ollie
17. Triple-Ollie

In order to make a Quadruple-Ollie possible, then, we should rewrite the Quadruple-a rule so that

1. Triple-a a

5. What other types of combinations of the listed combos can never actually be pulled off by the player, and why not?

There are a large (in fact, infinite) number of possible combos that the parser can never actually parse, due to the fact that it recognizes some sub-sequence of the combo as a different combo and reduces it, rendering that sub-sequence unusable by the original rule.
For example, you can never perform a Double-Nollie (or any further iteration of Nollies), because the parser recognizes a spurious Ollie inside of it:

\[ \text{x X S X S X S S X S X S S X S X S S X S} \text{Þ NOllie Ollie} \]

Likewise, any Inverted-Inverted-a, as well as anything built on it (like an Atomic-Inverted-a), will fail, because the parser always recognizes two consecutive 4s as a Crouch:

\[ \downarrow \downarrow \text{a} \uparrow \uparrow \text{Þ crouch a} \uparrow \uparrow \]

The same goes for any sort of Inversion of a Crouch or any move beginning in a Crouch, such as the Inverted-360-Kickflip. The first 4, which should be part of the Inversion part, is instead reduced along with the first 4 of the Crouch to make a spurious Crouch, and the leftovers are interpreted incorrectly as an Inverted-Backside-360:

\[ \downarrow \downarrow \text{a} \uparrow \uparrow \text{Þ crouch a} \uparrow \uparrow \downarrow \text{Backside-360} \uparrow \text{Þ crouch Inverted-Backside-360} \]
(H) LinearB

ko-no-so
a-mi-ni-so
pa-i-to
tu-ri-so
ku-do-ni-a
a-pa-ta-wa
ru-ki-to
u-ta-no
ku-pi-ri-jo
tu-ni-ja
2009 Solutions

(H) LinearB

a
do
i
ja
jo
ki
ko
ku
mi
ni
no
pa
pi
ri
ru
so
ta
to
tu
u
wa
2009 Solutions

(H) Linear B

H4

♀️♀️♀️ ‘girl’ ko-wa
♀️♀️♀️ ‘all’ pa-ta
♀️♀️♀️ ‘this’ to-so
♀️♀️♀️ ‘cumin’ ku-mi-no
♀️♀️♀️ ‘linen’ ri-no
In Bulgarian, the adjectives agree in number with the nouns they qualify. Thus, a noun in plural is accompanied by an adjective in plural, and a noun in singular is accompanied by an adjective in singular.

The formation of the plural of the nouns is complex and cannot be deduced from the data presented in this problem. The specific noun plural form is not relevant to the formation of the plural of the adjective which accompanies it.
2009 Solutions

(I) BgAdj

I. The problem presents only a partial picture of the formation of the plural of the adjectives. The three rules could be formulated in various ways, essentially equivalent to the following:

A. If the singular ends in –ен (stressed), then just add –i: červen : červeni. (15%)

B. If the adjective indicates from what matter the noun is made, then just add –i: kостен : kостени. (25%)

C. In all other cases, drop the final e and add –i: гневен : гневни. (10%)

II. In brackets, after each adjective, the rule symbol is given (this will vary according to the order in which the rules are listed!):

| 9 | obiknoven i (A) procedur i | ordinary procedures |
| 0 | ле с ni (C) уро ci | easy lessons |
| 2 | ри bni (C) re storant i | fish restaurants |
| 2 | кост ni (C) zabolyavani ya | bone disasters |
| 3 | лене ni (B) чарш аф i | linen sheets |

For each correctly formed plural, 10% of the full score awarded. Nothing accorded if error whatsoever; score awarded if there is misspelling irrelevant to the problem, e.g. obikoveni instead of obiknowneni.
## HypoHmongdriac

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>___ be lost</td>
</tr>
<tr>
<td>2</td>
<td>___ beef</td>
</tr>
<tr>
<td>3</td>
<td>___ beverage</td>
</tr>
<tr>
<td>4</td>
<td>___ bovine* livestock</td>
</tr>
<tr>
<td>5</td>
<td>___ chicken (the animal)</td>
</tr>
<tr>
<td>6</td>
<td>___ dog (the animal)</td>
</tr>
<tr>
<td>7</td>
<td>___ filthy animals; filth</td>
</tr>
<tr>
<td>8</td>
<td>___ filthy language</td>
</tr>
<tr>
<td>9</td>
<td>___ flesh; meat</td>
</tr>
<tr>
<td>10</td>
<td>___ hurt</td>
</tr>
<tr>
<td>11</td>
<td>___ internal organs; soul</td>
</tr>
<tr>
<td>12</td>
<td>___ language</td>
</tr>
<tr>
<td>13</td>
<td>___ liver (the organ)</td>
</tr>
<tr>
<td>14</td>
<td>___ livestock</td>
</tr>
<tr>
<td>15</td>
<td>___ lose heart (&quot;liver&quot;); lose one’s wits; panic</td>
</tr>
<tr>
<td>16</td>
<td>___ lose life to water; drown</td>
</tr>
<tr>
<td>17</td>
<td>___ lose money (&quot;silver&quot;)</td>
</tr>
<tr>
<td>18</td>
<td>___ lungs</td>
</tr>
<tr>
<td>19</td>
<td>___ money</td>
</tr>
<tr>
<td>20</td>
<td>___ small, non-bovine livestock</td>
</tr>
<tr>
<td>21</td>
<td>___ pig (the animal)</td>
</tr>
<tr>
<td>22</td>
<td>___ poetic genre (&quot;money-language&quot;)</td>
</tr>
<tr>
<td>23</td>
<td>___ silver</td>
</tr>
<tr>
<td>24</td>
<td>___ suffer from a headache (&quot;brain-ache&quot;)</td>
</tr>
<tr>
<td>25</td>
<td>___ suffer from grief (&quot;liver-ache&quot;)</td>
</tr>
<tr>
<td>26</td>
<td>___ suffer from lung disease (&quot;lung-ache&quot;)</td>
</tr>
<tr>
<td>27</td>
<td>___ water</td>
</tr>
<tr>
<td>28</td>
<td>___ water-buffalo liver</td>
</tr>
<tr>
<td>29</td>
<td>___ wealth</td>
</tr>
<tr>
<td>30</td>
<td>___ whisky</td>
</tr>
<tr>
<td>31</td>
<td>___ young female</td>
</tr>
<tr>
<td>32</td>
<td>___ young sow</td>
</tr>
</tbody>
</table>
(J) HypoHmongdriac

**Solution:**
28, be lost; 17, beef; 6, beverage; 15, bovine livestock; 13, chicken; 10, dog; 12, filthy animals; 23, filthy language; 18, flesh; 32, hurt; 3, internal organs; 24, language; 1, liver; 16 livestock; 25, lose heart; 27, lose life to water; 26, lose money; 2, lungs; 8, money; 14, small livestock; 11, pig; 22, poetic genre; 7, silver; 30, suffer from a headache; 29, suffer from grief; 31, suffer from lung disease; 4, water; 21, water-buffalo liver; 9, wealth; 5, whisky; 20, young female; 19, young sow

To solve this problem, it is important to realize that both of the two collections of words can be seen as networks, where words are connected by hyponymy relationships, and that these two networks must have equivalent shapes. However since “matching up” a whole network (or “graph”) of this kind with another is difficult even for a computer, solving this problem requires noting that the graphs are largely composed of smaller graphs with a tree-like shape. These are much simpler to deal with.
For example, you might observe that there are exactly two components of the graph where three words are hyponymns of a single word (like a tree with three branches) for both the Hmong and English collections. This allows you to infer that 25-28 and 29-32 must be either ‘be lost’ and the ‘lose’ words or ‘hurt’ and the ‘suffer’ words. You can determine how to match them by noting that only one of the roots in the Hmong words does not occur elsewhere (hlwb) and that only one of the English meanings does not occur elsewhere (‘brain’). This suggests that 29-32 must be the ‘hurt/suffer’ group and 25-28 must be the ‘lost/lose’ group. Furthermore, since sab occurs in both of these groups, and since ‘liver’ occurs in both groups, sab must be ‘liver’, sab-twm must be ‘water-buffalo liver’ and twm must mean ‘water buffalo’.

For example, you might observe that there are exactly two components of the graph where three words are hyponymns of a single word (like a tree with three branches) for both the Hmong and English collections. This allows you to infer that 25-28 and 29-32 must be either ‘be lost’ and the ‘lose’ words or ‘hurt’ and the ‘suffer’ words.
You can determine how to match them by noting that only one of the roots in the Hmong words does not occur elsewhere (hlwb) and that only one of the English meanings does not occur elsewhere ('brain'). This suggests that 29-32 must be the 'hurt/suffer' group and 25-28 must be the 'lost/lose' group. Furthermore, since sab occurs in both of these groups, and since 'liver' occurs in both groups, sab must be 'liver', sab-twm must be 'water-buffalo liver' and twm must mean 'water buffalo'.

This will lead you to the livestock tree in 12-16 and the realization that Hmong compounds are of at least two types. In one type, the meaning of the whole is the meaning of the first part modified by the second part (as in sab-twm). In the second type, the meaning of the whole is a general category including the meaning of both parts (that is, both parts are hyponyms of the whole). Knowing that twm is 'water buffalo' should allow you to guess that nyuj-twm is 'bovine livestock' since 'water buffalo' is a hyponym of only 'livestock' and 'bovine livestock', 'bovine livestock' is a hyponym of 'livestock' and nyuj-twm is a hyponym of qab-npua-nyuj-twm. We can now see that 3, 6, 9, and 12 are all compounds of the second type, and reason from what is known about their parts that 3 and 6 must be 'internal organs; soul' and 'beverage'. We see that 12 must be 'filthy animal; filth' since it occurs embedded inside of a type one compound that can only mean 'filthy language' (23). Therefore, 14 must be 'small, non-bovine livestock'.

By applying similar logic to the remaining cases, you will arrive at the answer given above.
Word orders:
- SOV – in case the subject is a pronoun
- OSV – in case the subject is not a pronoun

Pronouns:
- ŋinda = you
- ŋađa = I

Definite articles:
- bayi, placed before subjects of intransitive verbs and objects of transitive verbs
- baŋgul, placed before subjects of transitive verbs.

The suffix -ŋgu is placed on the subject of transitive verbs, when the subject is not a pronoun.

Assignment 1. Give the English translations:
- bayi ŋalŋga banagaŋu = The boy returned.
- bayi yaŗa baŋgul yuŗĩŋgu walmbin. = The kangaroo waked the man.
- ŋinda bayi yuŗi buŗan. = You saw the kangaroo.

Assignment 2. Give the Dyirbal translations:
- You sat. = ŋinda ŋinaŋu
- I caught the kangaroo. = ŋađa bayi yuŗi ŋiman.
- The father waked the man = bayi yaŗa baŋgul ŋumāŋgu walmbin.
2009 Solutions

(L) YakDuDray

L1. (16/25 points: two points for each correct match except for the blank)

1 2 3 4 5 6 7 8 9
D E H B C A G F _

Intermediate data:

A. Kuvi 8-2=6
B. Albanian 10-6=4
C. Farsi 8-3=5
D. Irish 6-5=1
E. Nepali 6-4=2
F. Yiddish 9-1=8
G. Pengo 10-3=7
H. Lithuanian 8-5=3

(5x4)+(9x8)=(5x10)+(6x7)

L2. (9/25 points)
Sample "key insights"

- closer connections among neighboring languages
- consonants more likely to be preserved
- pronunciation may not match spelling
- specific phonological changes, e.g., s-sh, c-p,
- specific patterns for numerals, e.g., 9 starts with N, 4 has T+R in the middle
- use of the title of the problem
- use of the equation
- use of the constraints imposed by the subtractions
- the form for the number 1 changes the most
M1. Here is the revised grammar. The changes are relatively small.

Sentence -> PosNounPhrase + Verb + NegNounPhrase
Sentence -> NegNounPhrase + Verb + PosNounPhrase

PosNounPhrase -> PosAdjective + Noun
PosNounPhrase -> PosAdjective + PosNounPhrase
NegNounPhrase -> NegAdjective + Noun
NegNounPhrase -> NegAdjective + NegNounPhrase

Noun -> people
Verb -> love
PosAdjective -> good
PosAdjective -> charming
PosAdjective -> happy
NegAdjective -> bad
NegAdjective -> obnoxious
NegAdjective -> unhappy

Notice that in this grammar, a single Noun does not qualify as a PosNounPhrase or NegNounPhrase. This ensures that the false statement "people love good people" is ungrammatical, since "people" is not a NegNounPhrase.

M2. Could it help to list 1-word bad phrases? No. You can't list any of the 8 vocabulary words without ruling out some legal sentences. (And there is no point in listing words outside that vocabulary, since they will have no effect and you were were asked to keep your list as short as possible.)

How about 2-word bad phrases? There are 25 types of 2-word phrases: the first word can be from any of the 5 categories [START, Noun, Verb, PosAdjective, NegAdjective], and the second word can be from any
of the categories \{Noun, Verb, PosAdjective, NegAdjective, END\}. Of these 25 types, the following 15 types can never appear in a legal sentence, so we list them as bad phrases:

\[
\begin{align*}
\text{START Noun} & \quad (1) \\
\text{START Verb} & \quad (1) \\
\text{START END} & \quad (1) \\
\text{Noun Noun} & \quad (1) \\
\text{Noun PosA} & \quad (3) \\
\text{Noun NegA} & \quad (3) \\
\text{Verb Noun} & \quad (1) \\
\text{Verb Verb} & \quad (1) \\
\text{Verb END} & \quad (1) \\
\text{PosA Verb} & \quad (3) \\
\text{PosA NegA} & \quad (9) \\
\text{PosA END} & \quad (3) \\
\text{NegA Verb} & \quad (3) \\
\text{NegA PosA} & \quad (9) \\
\text{NegA END} & \quad (3)
\end{align*}
\]

The *remaining* 10 types are depicted by the 10 arrows in this graph:
[insert bigram.png here]

By allowing only those 10 types of 2-word phrases, the device so far allows any sentence that corresponds to a path in the graph. Now, where does that leave us? As you can see, this already ensures that

* START must be followed by one or more Adjectives of the same type, and then a Noun. In other words, START must be followed by a PosNounPhrase or NegNounPhrase.
(M) Orwellspeak

* Such a PosNounPhrase or NegNounPhrase may be followed by END, or else may be followed by a Verb and another PosNounPhrase or NegNounPhrase.

However, this still permits illegal utterances like

A1. good people (not a sentence)
B1. good people love good people (not true)
C1. good people love bad people love good people (not a sentence)

and similarly

A2. good charming people
B2. good charming people love good charming people
C2. good charming people love bad obnoxious people love good charming people

We can get rid of some of the A. sentences with the 4-word bad phrases

    START PosA Noun END (3)
    START NegA Noun END (3)

This is only able to get rid of the shortest A. utterances, such as A1. We would need longer bad phrases to get rid of A2., since every 4-word subsequence of A2. can be part of a legal sentence. No finite list of bad phrases can get rid of all the A. utterances -- even with an upgraded device that allowed 1000-word bad phrases, we would not be able to censor extremely long A. utterances.

Similarly, we can get rid of some of the C. sentences with the 4-word bad phrases

Verb PosA Noun Verb (3)
2009 Solutions

(M) Orwellspeak

Verb NegA Noun Verb (3)

Again, this is only able to get rid of the shortest C. utterances, such as C1. We would need longer bad phrases to get rid of C2., and no finite list could get rid of all the C. utterances.

However, we can get rid of *all* of the B. utterances with only the 4-word bad phrases PosA Noun Verb PosA (9) NegA Noun Verb NegA (9)

These require successive noun phrases to be of opposite polarity. They work on noun phrases of *any* length, by requiring the first phrase's last adjective to oppose the second phrase's first adjective. For example, we are able to censor B2. because it contains "... charming people love good ..."

The total number of bad phrases above is 73.

M3. Yes. It fails to censor A2. and C2. above.

M4. A single 1-word bad phrase will satisfy the government's stated needs by censoring everything:

START
Or they could use
END

(Or if the device can handle allow 0-word bad phrases, then the single 0-word phrase ""
will also censor everything, as it is contained in any utterance; think about it!)
2009 Solutions

(M) Orwellspeak

You may be interested in some connections to computational linguistics:

* Problem M1 asked you to write a tiny context-free grammar. It is possible to write large context-free grammars that describe a great deal of English or another language. Although the "Opposites Attract" setting was whimsical, you could use similar techniques to ensure that plural noun phrases are not the subjects of singular verbs, and -- for many languages -- that plural noun phrases only contain plural adjectives.

* Problem M2 asked you to approximate the context-free grammar by what is called a 3rd-order Markov model, meaning that the model's opinion of the legality or probability of each word depends solely on the previous 3 words. (That is, the model only considers 4-word phrases.) The graph shown partway through the solution depicts a 1st-order Markov model (which considered only 2-word phrases).

* Problem M3 showed that the Markov model was only an approximation of the context-free grammar -- it did not define exactly the same set of legal sentences. The solution further noted that *no* nth-order Markov model could exactly match this context-free grammar, not even for every large n.

If you know about regular expressions, you may have noticed that the following regular expression *would* be equivalent to the context-free grammar, hence would do a perfect job of censorship.

```
START ( ((PosA)+ Noun Verb (NegA)+ Noun)
  | ((NegA)+ Noun Verb (PosA)+ Noun) ) END
```

Regular expressions or regular grammars are equivalent to finite-state machines. They are not as powerful as context-free grammars in
general, but they are powerful enough to match the "Opposites Attract" grammar. They are essentially equivalent to hidden Markov models, an important generalization of Markov models.

* Problems M3 and M4 together were intended to make you think about how to measure errors. In general, a system that tries to identify bad sentences (or bad poetry or email spam or interesting news stories) may make two kinds of errors: it may identify too many things or too few. Both kinds of errors are bad, and there is a tradeoff: you can generally reduce one kind at the expense of the other kind. The original requirement in problem M2 was to completely avoid the first type of error (i.e., never censor good stuff) while simultaneously trying to avoid the second type of error (censor as much bad stuff as possible). But the revised requirement in problem M4 considered only the second type of error, giving the vendor an incentive to design a dumb system that did horribly on the first type of error. You might conclude that when evaluating a vendor's system or setting requirements for it, you should pay attention to both kinds of error.