2009 Solutions

(M) Orwellspeak

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:

START Noun (1)
START Verb (1)
START END (1)

Noun Noun (1)
Noun PosA (3)
Noun NegA (3)

Verb Noun (1)
Verb Verb (1)
Verb END (1)

PosA Verb (3)
PosA NegA (9)
PosA END (3)

NegA Verb (3)
NegA PosA (9)
NegA END (3)

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)

(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!)

(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.

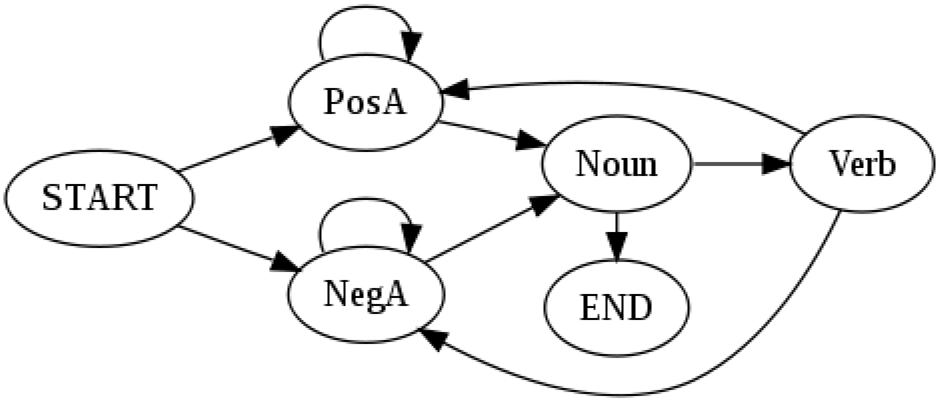
```
START (  ((PosA)+ Noun Verb (NegA)+ Noun)
     | ((NegA)+ Noun Verb (PosA)+ Noun)  ) END
```

Regular expressions or regular grammars are equivalent to finite-state machines. They are not as powerful as context-free grammars in

(M) Orwellspeak

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.


PosA
START
Noun
Verb
NegA
END