COSI135: Fall 2014
Assignment #4

This assignment is geared toward expanding the lexicon of the Haskell parser `P.hs` in preparation for the next programming assignment.

You are given a modified version of the `P.hs` parser that allows for handling of multiple verb tenses and aspects: the simple past ("John walked"), the simple present ("John walks"), the simple future ("John will walk"), and the present perfect ("John has walked"). Your task is to expand the verbs present in the lexicon into all of these forms.

You can use the following lists of common English verbs for reference:

- http://www.linguasorb.com/en/english/most-common-verbs
  (Green denotes a regular verb, red an irregular verb)
- http://www.acme2k.co.uk/acme/3starz%20verbs.htm
  (Bold denotes an irregular verb, red denotes a modal verb)

You should create inflection paradigms for approximately 100 verbs, and we recommend you avoid modal verbs and auxiliaries like "be," "do," "have," and "will." Even though they have non-auxiliary and non-modal sense, you will run into headaches when you try to formulate the semantics of these verbs in Haskell.

You should consult the Brandeis Verb Lexicon (http://pubpages.unh.edu / ˜jel/712/examples/verbinfo.html) for typing information on a large number of verbs. A key to the various codes can be found at the top of the file. You should cross-reference the verb you choose to implement with the Brandeis Verb Lexicon to determine the semantics of that verb. You can then examine some of the existing examples in `Lexicon.hs` to model your verb paradigms on. For instance "shout" is an intransitive verb, and so is "age" according to the BVL, so a paradigm for "age" could be modeled on that given for “shout.” Similarly, the BVL gives similar typing for “help” and “freeze,” so the latter could be modeled after the example of the former given in `Lexicon.hs`.

You should make your edits in the file Lexicon.hs. All your verb forms need to be inserted before the last line `lexicon _ = []`. You will see a sample inflection paradigm for *shout* near the bottom of the file. The easiest way will probably be to treat periphrastic constructions like the future and the perfect as single units, like `will_shout` and `has_shouted`, as shown in the file. This way you won't have to worry about handling “will” and “have” in those cases as auxiliaries.

(If you want to handle “will” and “have” as auxiliaries, you are free to do the extra work to do so by modifying `P.hs` – HINT: Examine how “did” and “didn't” are handled in `Lexicon.hs` and trace it through `P.hs`. This is **not** part of the assignment, but will earn extra credit.)

Due to the large number of verbs you will need to inflect and codify into the file, it is recommended that you process the verbs you choose with some kind of script. This can take any form you want, written in any language you want.

Your modified `Lexicon.hs` file should work with the existing parser. To test this, save your file and load `HRAS.hs` in GHCI. If HRAS successfully compiles all its dependent modules, your file is programatically well-formed. You should test this before you submit.

You should submit your modified parser suite (include all files), the list of verbs you added to the lexicon, and the script file you used to process the verbs into Haskell code (if you created one).