Lydia Simmons
Functional Programming - Project Proposal
16 May 2017

Overall description:
I was pretty taken with the natural-language parser demonstrated in class. My goal is to modify that system to work on Latin sentences, which will require some interesting algorithmic changes: in Latin, word order doesn't matter, but grammatical agreement between nouns and adjectives, verbs and objects, etc. is much more robust than in English and determines sentence structure. The algorithm to determine sentence structure will have to be very different for Latin. The typing system for words may or may not end up looking similar.

Initial work:
There is a handy tool from the Perseus Digital Library that will output raw XML for Latin word forms, like so:
http://www.perseus.tufts.edu/hopper/xmlmorph?lang=la&lookup=amat

```
<analyses>
  <analysis>
    <form lang="la">amat</form>
    <lemma>amo</lemma>
    <expandedForm>amat</expandedForm>
    <pos>verb</pos>
    <person>3rd</person>
    <number>sg</number>
    <tense>pres</tense>
    <mood>ind</mood>
    <voice>act</voice>
    <dialect/>
    <feature/>
  </analysis>
</analyses>
```

This tells us that "amat" is a third person singular present active indicative verb.
I've used the download and xml libraries to create a program that, given a Latin word, will hit that URL, parse the XML, and use the data to create an instance of a custom Word type with the appropriate attributes. (It only knows about nouns so far, but we're getting there.)

Milestones:

- Teach the program about other parts of speech.
- Write a simple algorithm that will parse a small subset of Latin grammar, probably noun/adjective agreement.
- Expand the algorithm as time permits to recognize grammatical constructs of various complexity. This might end up looking like a domain-specific language, like the one given for "find" in the current homework…
- The big challenge is to create an algorithm that doesn't take too long, since this could easily generate an intractable number of permutations.