Homework 1
Solutions

1. Exercise 1c and 1e in the exercises in section 2.7.3, pg 64 of Bird et al.

   c. numbers, either integer or real (with decimals)

   e. non-empty strings either of alphanumeric "he47lp" or punctuation ".,!" (actually any non-alphanumeric and non-whitespace). Strings must be one or the other, not a mix!

2. Exercise 2.2b in the exercises in section 2.7.3, pg 64 of Bird et al.

```
from nltk import re_show
import re
p = re.compile(r'\d+((\* | \+)\d+)*') # or r'\d+([*+]\d+)*'
print p.findall('23+47*13')
print "show"
re_show(r'\d+((\*|\+)\d+)*','23+47*13    34*3*4+5')
```

3. Exercise 4 in the exercises in section 2.7.3, pg 65 of Bird et al.

   A short solution (from Kris Karr):

```
def toPigLatin(word):
p = re.compile(r'([^aeiouAEIOU]*)(\w*)')
m = p.match(word)
first = m.group(1)
second = m.group(2)
print second + first + 'ay'
```

   and a longer one:

```
def piglatin(word):
begin = True
pre=''
letternum = 0
new = ''
while letternum < len(word) and begin:
   if begin and word[letternum] not in ['a','e','i','o','u']:
      pre+=word[letternum]
      letternum+=1
   else:
      begin=False
print(pre)
while letternum < len(word):
   new+=word[letternum]
   letternum+=1
```

```
    return new+pre+'ay'

    print piglatin('bill')
    print piglatin('arrive')
```

4. Exercise 2.1cdf in the exercises at the end of Chapter 2, pg 28 of Jurafsky and Martin. (But use python, not perl! Use "raw" strings to avoid problems with backslashes – see Bird, pg. 60, for details)

   c. `r'\b(\w+)\s\1\b'`

   d. `r'b(b|ab)*'`

   f. `r'.*((\bgrotto\b.*\braven\b)|(\braven\b.*\bgrotto\b)).*'`

5. Exercise 2.4 in the exercises at the end of Chapter 2, pg 29 of Jurafsky and Martin.

   Too hard to draw in Latex!

6. The file wsj15-18.txt is available on the CS computer system at `/common/cs/cs181/data/wsj15-18`. This comprises over a megabyte of text from the Wall Street Journal. Please answer the following questions by writing python programs and utilizing the nltk library. (Be sure to turn in your programs as well as the answers.)

   (a) Find all words that include all of the vowels (a, e, i, o, and u) in order. Words are strings separated by whitespace. In particular, hyphenated words count as a single word.

```
import re

wsj = open('wsj15-18.txt','rU').read()
wsj = wsj.lower()

p = re.compile(r'\b\S*a\S*e\S*i\S*o\S*u\S*\b')

print p.findall(wsj)
```

   Result is "marked-if-touched".

   (b) Separate the contents of the file into sentences and words.

      i. How many unique words are in the text?
      There were multiple results, ranging from 13,931 to 187,350(!!). The "correct" answer is likely about 17,000, but it depends on how you decide what a word is.

      ii. How many sentences are in the text.
      Answers ranged from 8270 to 13,829. Again the difficulty is how you decide what a sentence is. Numbers around 8900 were the most popular.

      iii. What is the average word-length of a sentence?
      Answers ranged from 11.725 to 23.99. Around 23 was the consensus choice.

      iv. Print out the shortest and longest sentences and report their lengths.
      There were several 2 word sentences (actually one word if you don't count punctuation), like "Nonsense!"

There were also several proposals for the longest sentence. These included "Shorn of all of their rules . . . ", "These proposals . . . ", "H & R Block . . . ", and "Treasury Bills . . . ". The first was most popular.

Hint: Test your programs on shorter excerpts from this text file before running it on the whole file.