Natural Language Processing
Lab 1 Solutions

This lab sheet is to practice the concepts taught so far, which is just regular expressions.

1. Write regular expressions for the following languages.

    1. The set of all alphabetic strings.

    > **Ans:** /[a-zA-Z]+/

    2. The set of all lowercase alphabetic strings ending in a b.

    > **Ans:** /[a-z]*b$/

    3. The set of all strings from the alphabet a, b such that each a is immediately preceded and followed by a b.

    > **Ans:** /(b+(ab)+b+)?/

2. Write regular expressions for the following languages. By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

    1. The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug"). You may use \s to match a whitespace character to make things clear.

    > **Ans:** /([a-zA-Z]+)\s+\1/

    2. All strings that start at the beginning of the line with an integer and that end at the end of the line with a word. You may use \b to match the empty string, but only when it is not at the beginning or end of a word.

    > **Ans:** /^[0-9]+[a-zA-Z]+$/

3. All strings that have both the word *grotto* and the word *raven* in them (but not, e.g., words like *grottos* that merely contain the word *grotto*).

   Ans: /\bgrotto\b.*\braven\b|\braven\b.*\bgrotto\b/

4. Write a pattern that places the first word of an English sentence in a register. Deal with punctuation.

   Ans: /^[^a-zA-Z]*([a-zA-Z]+)/

3. Implement an ELIZA-like program, using substitutions such as those described on page 10. You might want to choose a different domain than a Rogerian psychologist, although keep in mind that you would need a domain in which your program can legitimately engage in a lot of simple repetition.

Ans:

```
import re, string
patterns = [(r"\b(i'm|i am)\b", "YOU ARE"),
(r"\b(i|me)\b", "YOU"),
(r"\b(my)\b", "YOUR"),
(r"\b(well,?) ", ""),
(r".*YOU ARE (depressed|sad) .*",
r"I AM SORRY TO HEAR YOU ARE \1"),
(r".*YOU ARE (depressed|sad) .*",
r"WHY DO YOU THINK YOU ARE \1"),
(r".*all .*", "IN WHAT WAY"),
(r".*always .*",
"CAN YOU THINK OF A SPECIFIC EXAMPLE"),
(r"[%s]" % re.escape(string.punctuation), ""),]

while True:
    comment = input("User:")
    response = comment.lower()
    for pat, sub in patterns:
        response = re.sub(pat, sub, response)
        print(response.upper())
```