Homework #5: CMPT-413

Distributed on Mon, Mar 1; Due on Mon, Mar 8

Anoop Sarkar – anoop@cs.sfu.ca

(1) **Trees in Perl** (Submit file: `testTreeModule.pl`)

Get the Perl package `TreeModule.pm` from the location specified on the course web page. It encapsulates a variety of Perl functions that represent and manipulate trees.

Create a new Perl program called `testTreeModule.pl` to use this package to create and work with trees.

This is a snippet of Perl code that uses the TreeModule package to convert a tree into a Perl data structure using the `stringToTree` function. The tree stored in the reference `$tree` which can then be converted back to a string using the function `treeToString`.

```
use TreeModule;
use strict;

my $treeString = "(S (NP) (VP (V) (NP)))";
my $tree = TreeModule::stringToTree($treeString);
my $output = TreeModule::treeToString($tree);
print "$output\n";
```

Executing the above Perl code produces the following output:

```
(S (NP)
   (VP (V)
       (NP)))
```

Extend the above Perl code to do the following:

a. Create a new Perl tree for the following tree (let's call it `$pp`):

```
(PP (P)
    (NP))
```

Then add the above tree as the third child of the `VP` node in the original tree `$tree` (do not use the `stringToTree` function), creating a new tree that looks like this:

```
(S (NP)
   (VP (V)
       (NP)
       (PP (P)
           (NP))))
```

b. Use the above tree, plus the tree `$pp` to create the following tree (do not use the `stringToTree` function):

```
(S (NP (PP (P)
           (NP)))
   (VP (V)
       (NP)
       (PP (P)
           (NP))))
```

c. Use the function `treemap` in the package `TreeModule` to map each node in the tree to lowercase (do not use the `stringToTree` function). The output should look like:

```
(s (np (pp (p)
           (np)))
   (vp (v)
       (np)
       (pp (p)
           (np))))
```

d. Without using the function `stringToTree` using only the trees converted to Perl in the above steps, create a new Perl tree that looks like:

```
(PP (PP (P)
        (NP))
    (PP (P)
        (NP)))
```

Write down all the above steps in a single Perl file, which when executed will print out each of the trees shown above using the `treeToString` function in the package `TreeModule`.

The pretty indented trees can be created by running the output of your file through another Perl program called `indentrees.pl` by running the command:

```
perl testTreeModule.pl | perl indentrees.pl
```

`indentrees.pl` is available from the same directory as `TreeModule.pm`.