A1: Tree Enumeration

Purpose of Assignment:

- Generate all rooted, ordered trees with ***n*** nodes, and output them in “parenthesized” form
- Compare and contrast the programming experience in each of the different languages (Ada, C#, Prolog, Python, OCaml)
    - What was easy/hard?
    - Are there noticeable differences in speed?
    - What’s there to like/dislike?

Approach:

The method that made the most sense to me was to generate the tree using a “depth first search” like approach. I chose to create a procedure that would be recursively called. In each call of this procedure, it would choose to take every possible action within the parameters it was given (Number of nodes left & Depth). This in itself produced a tree like structure of all possible rooted, ordered trees with ***n*** nodes. On each leaf of this tree is essentially one of those enumerated trees with ***n*** nodes. Once understanding this, it was trivial process outputting the resulting leaves of the tree into “parenthesized” form.

Something I found interesting, was that the number of trees generated from this method, followed the *Catalan Numbers*. They are a sequence of natural numbers that occur in various counting problems, often involving recursively-defined objects. My trees being produced recursively could be one reason why these two things correlate.

Jean-Marc Boullianne
jboullia@u.rochester.edu

Comparing Languages:

The hardest part about this project was having to switch your mindset around for the different languages. The easiest language to code this program in was prolog, as it lends itself to the recursive approach. On the other hand, I had the hardest time with ada as I didn't know the language, and because of it's difficulty concatenating strings (Had to instead use unbounded strings). A pro for ada though is that it is seemed to be much faster than the other languages I used.

On another note, I liked the simplicity for coding a recursive based procedure in prolog. It's natural and fluid. The other languages tend to follow a similar approach to each other, which made things easier in porting the program from language to language.

How to run: ("~" indicates command line prompt)

Ada:

```
~ gnatmake treeenumeration.adb -o TreeEnumeration
~ ./TreeEnumeration <NumberOfNodes>
```

C#:

```
~ mcs TreeEnumeration.cs
~ mono TreeEnumeration.exe <NumberOfNodes>
```

Prolog:

```
~ swipl TreeEnumeration.pl
~ enumerateTrees(<NumberOfNodes>).
```

Python:

```
~ python3 TreeEnumeration.py <NumberOfNodes>
```

OCaml:

```
~ ocaml TreeEnumeration.ml <NumberOfNodes>
```

Jean-Marc Boullianne
jboullia@u.rochester.edu

Included Files:

- `TreeEnumeration.py`
- `TreeEnumeration.pl`
- `TreeEnumeration.ml`
- `TreeEnumeration.cs`
- `treeenumeration.adb`