Expression Trees

Cpt S 321

Washington State University

Expression Trees Needed for Spreadsheets

- Some sort of Binary Trees
- Used to parse and calculate the formulas in the spreadsheet cells
- Currently, our formulas don’t really need them
  - For HW4, we can only set one cell equal to another
  - But for later homehoworks the following operators are required: +,-,*, /
  - Also, more cell references and complex sub-operations will be included

Expression Trees Needed for Spreadsheets

- Tree nodes need to be an abstract class with specialized sub-types:
  - Node representing a constant numerical value
  - Node representing a variable
  - Node representing a binary operator

Things to keep in mind

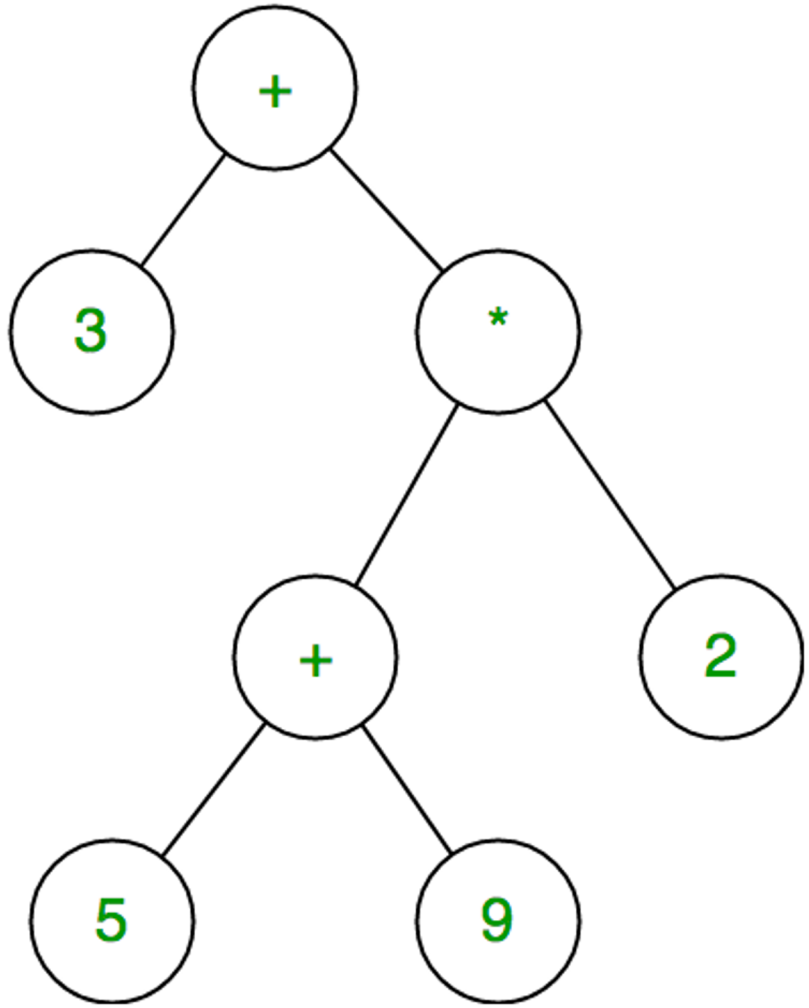
- Binary Expression Trees ≠ General Binary Trees:
  - A node in a binary tree can have at most two children, right and left
  - In this assignment, a node (N) in our trees have either zero or two children

- If N has zero children what kinds of nodes are possible for N?
  - Number: string to double conversion to evaluate
  - Variable: dictionary lookup on key to return a string number

- If N has two children, what kinds of node(s) are possible for N?
  - Operand (with two children --- no unary operators for now…)

Simple Expression Tree

- Start at root
- Root can only operate on values
- Recursively evaluate children if not values
  - Result will be values (in this case)
- What is the expression that corresponds to this tree?



Examples of expressions

1. 3+5+4
2. 3-2+1
3. 10/(7-2)
4. 10/(2*5)
5. (((((2+3)-(4+5)))))

Questions that we need to think about?

- What are the classes that we need and how are they connected?
- How do we construct an expression tree given an expression?
- Operators:
  - Are all operators the same? (Not a problem for HW5 but will become a problem later)
  - How do we design for change?

Expression Tree Code Demo

- Let's look at an example of how **NOT** to implement this
- Tasks for today:
  1. Does that even work?
  2. Find **problems** and **solutions** in the code and write this in the survey
     - **Hint**: 1 solution might solve more than one problem
     - Use bullet points and be brief.

The “out” modifier

- Similar to “ref” but
- For the “out” modifier:
  - Variables need not be initialized before going into the function
  - Must be assigned before going out of the function

```
class Test {
    static void SplitName(string fullName,
                          out string firstNames,
                          out string lastName)
    {
        int i = name.LastIndexOf (' ‘);
        firstNames = name.Substring (0, i);
        lastName = name.Substring (i + 1);
    }

    static void Main() {
        // a and b are declared on the fly:
        SplitName("Stevie Ray Vaughan",
                  out string firstName,
                  out string lastName);
        Console.WriteLine (firstName); // Stevie Ray
        Console.WriteLine (lastName); // Vaughan
    }
}
```