

Java Methods Implementation

Aidan Chin, Rayan Azad, Anhad Singh, Seth Oswalt

April 5, 2024

Writing Exercise

Write (in real Java) the following methods to be added to this class:

```
public class BooleanExpression {
    public static final int AND = 1; // AND operator
    public static final int OR = 2; // OR operator
    public static final int NOT = 3; // NOT operator

    boolean isLeaf; // true if this is a one-node tree
    boolean leafValue; // value of tree if it has just one node
    int operator; // value must be AND, OR, or NOT
    BooleanExpression left, right; // subexpressions if !isLeaf

    public BooleanExpression (boolean arg) {
        // create one-node tree with given value
        isLeaf = true;
        leafValue = arg;
        left = right = null;}

    public BooleanExpression (int op, BooleanExpression leftArg,
                             BooleanExpression rightArg) {
        // create tree with given operator, left argument, right
        // argument
        isLeaf = false;
        operator = op;
        left = leftArg;
        right = rightArg;}

    public boolean getIsLeaf () {return isLeaf;}
    public boolean getLeafValue ( ) {return leafValue;}
    public int getOperator () {return operator;}
    public BooleanExpression getLeft () {return left;}
    public BooleanExpression getRight () {return right;}}
```

1. A method `size` that returns an `int` giving the number of nodes in the calling expression's tree. (We'll mostly do this on the blackboard.)

2. A method `leaves` that returns an `int` giving the number of leaves in the calling expression's tree.
3. A method `depth` that returns an `int` giving the depth of the tree, which is the number of nodes in the longest directed path from the root node to any leaf.
4. A method `eval` to return the boolean value of the calling expression.
5. (if time) A method `toString` that returns a `String` representing the expression, in an infix format like "NOT ((true OR false) AND (false OR true))".

Response Area

```
class JavaMethods:
    public int size():
    {
        if (getIsLeaf()) return 1;
        if (getOperator() == NOT) return 1 + getLeft.size();
        return 1 + getLeft.size() + getRight.size();
    }

    public int leaves():
    {
        if (getIsLeaf()) return 1;
        return getLeft.leaves() + getRight.leaves();
    }

    public int depth(self):
    {
        if (getIsLeaf()) return 0;
        int leftDepth = getLeft.depth();
        int rightDepth = getRight.depth();
        return max(leftDepth, rightDepth);
    }

    public boolean eval(self):
    {

    }

    public str toString(self):
        # Write your Java code for toString method here
        pass
```
