CSCE625: Artificial Intelligence
Programming Assignment 3 : Simplifying Mathematical Expressions via Search

Name: Satya Sreenadh Meesala
UIN: 232003265.

SEARCH ALGORITHM:

I have used A* - Search Algorithms with g(n) = 0 and h(n) as mentioned below.

1) simplify
   Simplifies a given operation.
   For example : (Before : x + 3 = 3 + 4, After : x + 3 = 7)

2) solveIdentities
   Solves (sin(x)^2 + cos(x)^2 and cos(x)^2 + sin(x)^2)
   For example : (Before : sin(x)^2 + cos(x)^2 + y = 2 , After : 1 + y = 2)

3) squarIt
   Solves (sqrt(x) = equation)
   For example : (Before : sqrt(x) = 2 + 3 , After : x = (2 + 3) ^ 2)

4) unLogIt
   Solves (log(x) = equation)
   For example : (Before : log(x) = 2 + 3 , After : x = 10 ^ (2 + 3))

5) unLnIt
   Solves (ln(x) = equation)
   For example : (Before : ln(x) = 2 + 3 , After : x = e ^ (2 + 3))

6) inverseIdentity
   It takes the inverse of an operation and takes the operand to the other side of the equation. For example : (Before : x + 3 = 4 , After : x = 4 - 3)

7) commutative
   Gives commutative of two operands
   For example : (Before : x + 3 = 4 , After : 3 + x = 4)

HEURISTIC:

h = (2*findOperations(x) + findDepthOfX(x,self.variable) + ifXInLeft(x,self.variable) + ifXAtLeft(x,self.variable) + ifIdentityLeft(x))

where,
findOperations(x) : Function to find out number of operations left in equation
findDepthOfX(x, v) : Function to find depth of variable v
ifXInLeft(x, v) : Function to check if v in left subtree
ifXAtLeft(x, v) : Function to check if v is left child of root
ifIdentityLeft(x) : Function to check if identities left in the tree.

Rationale behind this heuristic:
1) It is good to have less number of operations in the equation to reach a solution faster hence “findOperations(x)” is being used. I have multiplied it by 2 to give it more priority.

2) We need variable ‘v’ at less depth in the tree to get it solved, so “findDepthOfX(x, v)” is being used.

3) We need variable ‘v’ in the left subtree of ‘=‘ in the equation. Hence “ifXInLeft(x, v)” is being used.

4) We need the variable ‘v’ on the left side of ‘=‘ in the equation. Hence “ifXAtLeft(x, v)” is being used.

5) We don’t want the identity sin^2 + cos^2 in the equation. Hence “ifIdentityLeft(x)” is being used.

So the node in the frontier with the minimum value of the combination of above mentioned things will be selected next.

OUTPUTS:

1) eq>x=(2+10)*(2^2)
    var>x
    Produces Output: x = 48

2) eq>x=(6*2)/(1-1)
    var>x
    Produces output: x = undefined

3) eq>2*sqrt(x)*3-y=pi
    var>x

```
        Produces Output: x=(((pi + y) / 3) / 2)^2



4) eq>2*x*3*y*4*z*5*6=800
        var>x
        Produces Output: x = ((1.11111111111 / z) / y)


5) eq>x=sin(y)^2+cos(y)^2+z
        var>x
        Produces Output: x = (z + 1)

6) eq>sin(x)^2 + cos(x)^2 + x + 2 = 5
        var>x
        Produces Output: x = 2

7) eq>sqrt(x)=(1+3-2)*(2^2)
        var>x
        Produces Output: x = 64

8) eq>sqrt(x)=(1+3-2)*(2^2) + sqrt(y)
        var>y
        Produces Output: y = (sqrt(x) - 8)^2

9) eq>log(x)+10*2-15=sin(x)^2+cos(x)^2+10
        var>x
        Produces Output: x = 1000000

10) eq>ln(x)*2=2*3*4/6
        var>x
        Produces Output: x = e^2

11) eq>sin(x)^2 + cos(x)^2 + ln(y) + 2 = 4 + log(x)
        var>y
        Produces Output: y = e^(((4 + log(x)) - 2) - 1)
```

RUNNING THE PROGRAM:

1) Run main.py
2) Input the equation after the prompt “eq>”
3) Input the variable for which the equation needs to be solved for after prompt “var>”

BRIEF NOTES AND LIMITATIONS

1) Used python 2.7.5
2) Multiple instances of the variable for which equation needs to be solved is not handled (eg. : “2x + 3 = 4 + x“ this is not handled)
3) Equations inside sprt/log/ln is not handled (eg: “sqrt(x+1) = 2” is not handled but “sqrt(x) = 2” is handled)
4) The identities “sin^2 + cos^2” will get solved to ‘1’ only if placed next to each other (eg. : “sin(x)^2 + cos(x)^2” is handled but “sin(x)^2 + y + cos(x)^2” is not handled)
5) Unary Minus is not handled
6) Changed color for WHITE to '' instead of '\033[97m'
7) Many helper Functions are used and all helper functions have their role mentioned above the definition
8) Calculus is not handled
9) e^x = 1 => x = ln(1) is not handled but ln(x) = 1 => x = e^1 is handled.

USED RESOURCES:

1) https://code.google.com/archive/p/aima-python/
2) http://robotics.cs.tamu.edu/dshell/cs625/asgn3/equationparser-0.1.tar.gz
3) https://docs.python.org for finding out usage of inbuilt libraries like operators.