

## **DESCRIPTION OF SEARCH ALGORITHM:**

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

## **ACTIONS:**

- 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  $(\text{sqrt}(x) = \text{equation})$   
For example : (Before :  $\text{sqrt}(x) = 2 + 3$  , After :  $x = (2 + 3)^2$ )
- 4) unLogIt  
Solves  $(\log(x) = \text{equation})$   
For example : (Before :  $\log(x) = 2 + 3$  , After :  $x = 10^{(2 + 3)}$ )
- 5) unLnIt  
Solves  $(\ln(x) = \text{equation})$   
For example : (Before :  $\ln(x) = 2 + 3$  , After :  $x = e^{(2 + 3)}$ )
- 6) inverseIdentity  
It takes 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:**

$$h = ( 2 * \text{findOperations}(x) + \text{findDepthOfX}(x, \text{self.variable}) + \text{ifXInLeft}(x, \text{self.variable}) + \text{ifXAtLeft}(x, \text{self.variable}) + \text{ifIdentityLeft}(x) )$$

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 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 variable ‘`v`’ in just 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 minimum value of the combination of above mentioned things will be selected next.