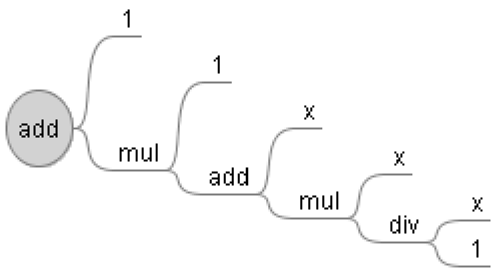
GP TEST ONE 2010-08-25

Objective:	Find program whose output matches x^2+x+1 over the range -1 <= x <= 1.
Function Set:	+, -, *, % (protected division).
Terminal Set:	X, and 1
Fitness:	Sum for absolute errors of x (-1.0, -0.9, ...0.9, 1.0)
Selection:	Fitness proportionate non elitist.
Initial Population:	Ramped
Parameters:	Population size: 100
Termination:	100 generations.
Best Individual:	**Run One:**



gp_add(1, gp_mul(1, gp_add(x, gp_mul(x, gp_div(x, 1)))))

(1 + (1 * (x + (x * (x / 1)))))
Simplifies To:
(1 + (x + (x * x)))

Run Two:
gp_add(gp_mul(gp_add(x, 1), x), 1)
(((x + 1) * x) + 1)

Run Three:
/**Same as Run One**/

Run Four:
gp_add(1, gp_add(x, gp_mul(x, x)))
(1 + (x + (x * x)))

Run Five:
gp_add(gp_mul(1, gp_mul(gp_add(1, x), x)), 1)
((1 * ((1 * x) * x)) + 1)

Run Six:
gp_add(gp_mul(x, gp_add(gp_mul(x, 1), 1)), 1)
((x * ((x * 1) + 1)) + 1)

Code:

```
from pyevolve import *
import math

error_accum = Util.ErrorAccumulator()

def gp_add(a, b):
   return a+b
def gp_sub(a,b):
   return a-b
def gp_mul(a, b):
   return a*b
def gp_div(a,b):
   '''
   "Safe" division, if divide by 0, return 1.
   '''
   if b == 0:
      return 1.0
   else:
      return a/(b*1.0)
def rangef(min, max, step):
   result = []
   while 1:
      result.append(min)
      min = min+step
      if min>=max:
         break
   return result

def eval_func(chromosome):
   global error_accum
   error_accum.reset()
   code_comp = chromosome.getCompiledCode()
   for x in rangef(-1, 1, .1):
      evaluated = eval(code_comp)
      target = x**2 + x + 1
      error_accum += (target, evaluated)

   return error_accum.getRMSE()

def main_run():
   genome = GTree.GTreeGP()
   genome.setParams(max_depth=5, method="ramped")
   genome.evaluator.set(eval_func)
   ga = GSimpleGA.GSimpleGA(genome)

   ga.setParams(gp_terminals = ['x', '1'],
                gp_function_prefix = "gp")

   ga.setMinimax(Consts.minimaxType["minimize"])
   ga.setGenerations(100)
   ga.setMutationRate(0.08)
   ga.setCrossoverRate(1.0)
   ga.setPopulationSize(100)
   ga.evolve(freq_stats=5)
   print ga.bestIndividual()
if __name__ == "__main__":
   main_run()
```

Joseph Lewis <joehms22@gmail.com>