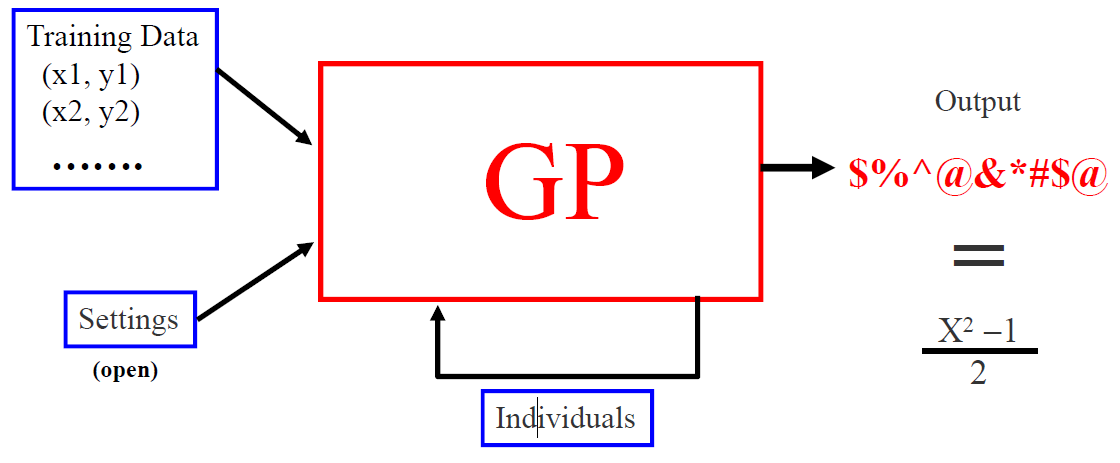
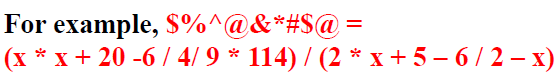
**Genetic Programming Project Requirements**

**Product Name: GP Genie**

**October 25, 2013**

**Diagram of Project Goal**





**Table of Contents**

[**1.0** **Introduction. 3**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624492)

[**2.0** **Purpose. 3**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624493)

[**3.0** **Scope. 3**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624494)

[**4.0** **Applicable Documents, and Regulations. 3**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624495)

[**5.0** **Definitions, Acronyms, and Abbreviations. 4**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624496)

[**6.0** **General Description. 5**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624497)

[**7.0** **Specific Requirements. 7**](https://docs.google.com/document/d/1hJhUsgxwm-MY0pq72AyT-df2WFLxf1kD0R9vzszOVZc/edit?usp=drive_web#_Toc346624498)

From IEEE standard 830 for software requirements specification (SRS) (book pg 197)

1. Introduction to the Document
   1. Purpose of the Product
      1. Automatically generate a function that is equivalent to the solution
   2. Scope of the Product
      1. Need some statement to bound our efforts and user expectations
   3. Acronyms, Abbreviations, Definitions
   4. References
   5. Outline of the rest of the SRS
2. General Description of the Product
   1. Context of the Product
   2. Product Functions
   3. User Characteristics
   4. Constraints
   5. Assumptions and Dependencies
3. Specific Requirements
   1. External Interface Requirements
      1. User Interfaces
         1. GUI Not Required
         2. **“Program shall obtain initial input configuration values from the user.”**
         3. **“Program will report periodic output data to the user while the program runs including, but not limited to:**
            1. Current generation being worked on
         4. **“Program will immediately conclude when a successful termination is reached and report the matching function to the user on screen.”**
         5. **“Program will conclude after fifteen minutes has elapsed and display (what?):”**
      2. Hardware Interfaces
      3. Software Interfaces
      4. Communications Interfaces
   2. Functional Requirements
      1. Configuration Data Input

**“Program shall support the following input data as variable configuration settings:”**

* + - 1. Number of candidates in population (integer, like 500)
      2. Depth of candidate tree (integer, like 30)
         1. How do we decide? - >can we systemize this?
         2. Max depth is low - like 5? -
         3. What is optimum number? Mathematical decision making or ask customer
      3. Fitness bar (percent, like 60%)
      4. *Crossover level? Or random generated…*
      5. *Mutation level? Or random generated…*
      6. *Input: amount of time?*
      7. *Get it started and start plugging in numbers to detect patterns*
         1. *collect data to analyze for emerging patterns*
    1. Training Data

**“Program shall generate? a set of training data consisting of no more than 30 positive and negative integers that vary in range from each other and covering a total range of 200… .”**

Do we feed the program the training data **(X’s)**? Generate it by rules for size, spacing? Hard code a list?

* + 1. Initialize Random Population

**“Program shall randomly generate a configurable number of populations of individual candidates for fitness evaluation.”**

**“Program will generate populations until a solution match is found or until 15 minutes has passed, whichever comes first.”**

**“Program shall user user input to determine the size of all the populations for that training execution. ”**

* + - 1. Individual Equation Candidate Creation

**“Program shall generate populations’ individual equations using a tree building method.”**

**“Program shall not allow trees to exceed a depth greater than a user configurable number.”**

**“Program shall randomly generate each tree node from a finite list of operands (0-9) and operators (+,-,/,\*).”**

**“Program shall prevent errors of bad operations (divide by zero) and bad configuration inputs (-300 generations).”**

* + - * 1. Random Select Node (Operator/Operand)
        2. Node selection is random until the max level minus one is reached
        3. At max level minus one, only operands may be selected

If Node is not a NaN(+-/\*), Select two more Nodes

Operand(0-9)

* + 1. Iterate over Candidates determining Fitness Evaluation

**“Program shall evaluate the fitness, to match equation Y = (X^2 - 1) / 2), of each individual in each population using provided training data.”**

* + - 1. Calculate Y Result Computation
      2. Calculate Training Data Result Computation (Y = (X^2 - 1) / 2)
      3. Calculate Error Delta
      4. Calculate Fitness of each Candidate
    1. Compare for Best

**“Program shall immediately approve of any individual matching the target equation exactly.”**

**“When a population doesn’t produce an approved equation, the program shall modify the population to create a new population using Selection, Reproduction, and Mutation algorithms.”**

* + - 1. **Natural Selection** - Select two Parents according to fitness

**“Program shall perform natural selection by using the configuration setting for a population’s fitness bar to eliminate the bottom unfit percent of the population based on their lack of fitness.”**

* + - * 1. Base selection on better fitness (better fitness, better chance of selection)

‘Bar’ for fit versus non-fit candidates is configurable

‘Bar’ will be reflected as a percent of the total population

* + - 1. **Reproduction** - Crossover Parents to produce new solution

**“Program shall perform reproduction by a method of crossover - swapping portions of their trees at a random level to produce two new solutions and evaluating their fitness.”**

* + - * 1. Randomly select a level in the Tree A
        2. Randomly select a level in Tree B
        3. Swap the two nodes at that level
        4. run
      1. **Mutation -**  Solution undergoes spontaneous changes (with small probability)

**“Program shall perform mutation by a method of changing out an element of their tree - either an operator or operand at some level - and evaluating its fitness.”**

* + - 1. (slide 5)Determine levels for crossover
    1. Mutate new solutions based on mutation probability
  1. Performance Requirements
  2. Design Constraints
     1. Program must be automated in order to handle requirements change
  3. Quality Requirements
  4. Other Requirements

1. Appendices