Submission Packet:
Charles Peabody / SEITZER

2014 Student-Faculty Collaborative Scholarship Program

Name of Student Proposer:

CHARLES PEABODY, Computer Science A & S, Class of 2015

Name of Faculty Advisor:

DR. JENNIFER SEITZER, Visiting Associate Professor (2013-2014), Associate Professor (2014-2015); Computer Science

Submitted by:

DR. JENNIFER SEITZER, Visiting Associate Professor (2013-2014), Associate Professor (2014-2015); Computer Science

Date Submitted:

Thursday, February 27, 2014

Checklist of Items Enclosed in this Packet (in order of Appearance):

1. Faculty-signed cover letter
2. Student written proposal
   a. Non-technical proposal summary
   b. Comprehensive literature review
   c. Description of the proposed work and the expected outcomes
   d. Project timeline
   e. Budget
   f. Reference page
3. 1-2 paragraph personal statement
4. Student-Faculty contract

ROLLINS

COMPUTER SCIENCE – ROLLINS COLLEGE – 1000 HOLT AVE., WINTER PARK, FL 32789 – T. 407-646-2432 – F. 407-646-2479

February 27, 2014

Professional Standards Committee
Rollins College
1000 Holt Ave.
Winter Park, FL 32789

Dear Colleagues:

I write this letter on behalf of Charles Peabody who is applying for a Student-Faculty Collaborative Scholarship (SFCS) for the period of June 23, 2014 -- August 15, 2014 to work with me in evolutionary computing (i.e., a sub-discipline of artificial intelligence) on a project entitled *Grammatical Evolution of a Robot Controller (called "GEF").* In this project, we will develop a software system that will use techniques of genetic algorithms, grammatical evolution, and robotics. The system, itself, will create a computer program to control a Finch robot. Thus, we are writing a computer program to output a different computer program.

Grammatical evolution is an iterative technique that repeatedly *creates* a set of possible solutions, *measures* each solution's efficacy with respect to a specified goal, and *selects* a few of the current solutions to generate the next set (in the next iteration). Each of our system's "possible solutions" is in the form of a robot control program; GEF will iteratively evolve an optimal robot control program to enable the robot to perform simple tasks such as obstacle avoidance and exiting a room in the dark.

Charles Peabody, the PI of this proposal, is an extremely diligent, respectful, and conscientious person. He was a student in two of my classes in Fall 2013 (in which he earned As) and is currently enrolled in two of my courses. He is a good student who consistently turns in beautiful, complete, and correct assignments. He is very mature and has an impressive history including serving eight years in the United States Marine Corps. To me, he seems to possess an inner strength that will serve him well in his current academic direction which includes graduate school.

This year (2013-2014), I feel fortunate to be serving as a Visiting Associate Professor at Rollins. In November 2013, I was offered a tenure-track associate professorship for the 2014-2015 academic year – which I happily accepted. I am thrilled to be re-starting my research program. As part of this effort, I formed a small research group in which Charles is a member, related to the SFCS proposal process. As a group, we met at least once a week, shared a Blackboard site, and chipped away at writing research proposals. As my students found out, proposal writing is hard work – only two out of five remain. Charles was always in attendance and was always prepared with the weekly assignments. He has built many robots as a hobby and has already started programming the robot simulator that we will be using during the evolution process. Moreover, I believe he has written an excellent proposal where he has clearly delineated the expected steps in this pursuit.

When programming a robot, one typically writes the computer code of the controlling program directly. Our research system, however, will *automatically* do this. This level of artificial intelligence is complex, difficult, and exciting. Because Charles has experience with direct programming of robots, using grammatical evolution will significantly expand Charles' repertoire and perspective as a computer scientist on how robots operate and the possibilities that exist.

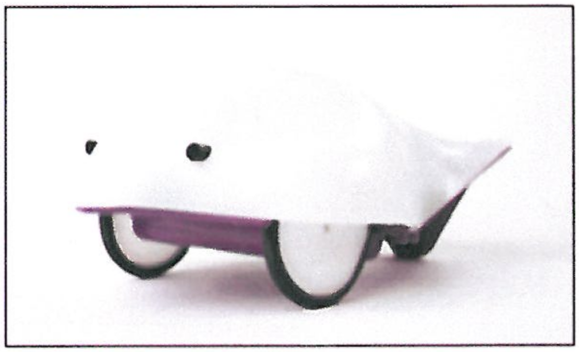
The components of *GEF* include the genome generator (a binary string), the phenome generator (the computer program created from the binary string), the simulator (to measure the newly generated program's efficacy), and the selector (to choose a few genomes to be used in the next iteration). To build this complex computer system, we will perform the following steps: 1) specify the system in pseudocode and diagrams, 2) define the total language that will be used for the generated robot control programs, 3) write the GEF software, 4) perform extensive executions (thereby creating many possible robot control programs), and 5) load the chosen programs to test on the actual robot: a Finch robot, developed at Carnegie-Mellon University. The Finch robot is particularly suitable for this project because it was developed to serve as a computer science educational tool to enable students to learn computer programming. Therefore, the Finch's controlling programs are relatively simple. Since GEF will be automatically generating these programs, simplicity is highly desired – at least in these early stages of the research.

Doing research is an act of great optimism. It is publicly affirming that one is willing to fail in order to succeed. Computer programming is a meta-level process, in that when we struggle over *how* to write a computer program to solve a problem, we are *problem solving about problem solving.* By embarking on this project to program this highly complex system of grammatical evolution, Charles Peabody is optimistically embracing a higher level of scholarship than normally experienced in the classroom. He is a good, honorable student who has great experience in both robotics and programming. I recommend him to you enthusiastically.

Very truly yours,

Dr. Jennifer Seitzer, Visiting Associate Professor (2013-2014)
continuing on as Associate Professor (2014- 2015)
Computer Science Department
Rollins College
407-646-2303

Grammatical Evolution of a Robot Controller



Charles Peabody, A&S

Department of Computer Science
Rollins College Class of 2015

Faculty Advisor: Dr. Jennifer Seitzer

Rollins College Student-Faculty Collaborative Scholarship Program
2/27/2014

I. Non-technical proposal summary

Artificial intelligence (AI) is a branch of computer science that develops machines and software that exhibit intelligence. Machine learning (an area of AI) involves building systems to extract patterns and knowledge from data to solve complex tasks [Russel2010]. One method of machine learning is the use of Genetic Algorithms (GA) inspired by mechanisms of biological evolution [Booker1989]. We plan to do research using a derivative of GAs known as Grammatical Evolution (GE). A GE computer system is one that automatically generates computer programs in any language [Ryan1998]. Therefore, the robot running a GE system can literally program itself. The work proposed here is to build such a GE system.

Work accomplished by the PI so far in preparation for this research includes the construction of several robots, active participation in a student research group, and intense reading on the material of Grammatical Evolution. We plan to continue collaboration through the remainder of Spring 2014 and Maymester 2014 so that we will be able to make optimal use of time and resources during participation in the Student-Faculty Collaborative Scholarship Program (SFCS) if the proposal is successful.

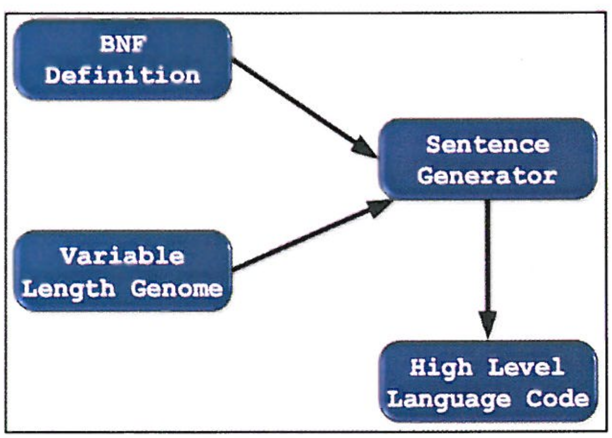
During the funded period of research, from June 23, 2014 through Aug 15, 2014, the researchers plan to create a GE system by doing the following: 1) program the genome generator to create each iteration's chromosomes, 2) program the phoneme generator (i.e., the evolving robot controller programs in progress), 3) build the robot simulator to facilitate testing of the programs in progress, and 4) connect these parts into one central GE system. When controller programs of moderate success have been created we will deploy them onto the Finch robot itself for further testing and refinement. In summary, we are requesting funding for student and faculty stipends ($4800), one Finch Robot ($100), and expected conference expenses ($1700).

II. Introduction

An autonomous robot uses on-board controllers to perceive, decide, and act on their environment [Russell]. There is a growing demand for autonomous robots to act intelligently in dynamic environments. Our proposed research in Grammatical Evolution (GE) is to create a system that uses techniques of Genetic Algorithms to create an optimal robot-controlling program to be executed by the robot controller.

Grammatical Evolution (GE) is a computational technique based on formal grammars. A formal grammar specifies how to make valid programming statements in a high level programming language such as Java. In GE, the high level programming language is represented in Backus-Naur Form (BNF) and uses the multiple options provided in BNF to exhaustively create the many different possibilities of programming statements.

Figure 1: The Grammatical Evolution Method



BNF notation expresses the grammar of a language using four core components [Warford]. These components can be expressed in the tuple {N, T, P, S}:

N: A set of non-terminal symbols that categorize a sub-phrase of the language.

T: A set of terminal symbols which makes up the language.

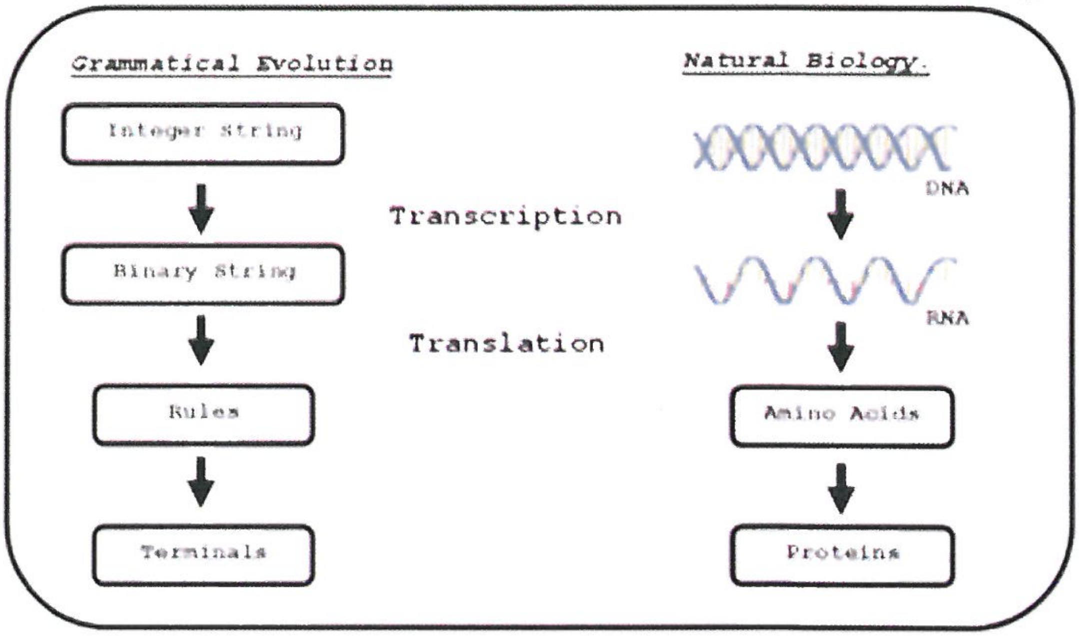
P: A set of production rules used to change non-terminal symbols into terminal symbols.

S: Start Symbol denoting the complete set of strings in a language.

GE uses the representational power of formal grammars with genotype-phenotype mapping principles from molecular biology. GE systems are iterative and execute a cycle of steps, which runs until it has met the stopping criteria set forth by the programmer. Starting with a genomic generator, GE produces variable length chromosomes consisting of 8-bit binary numbers. Each chromosome or, "individual", will derive a phenotype (which is a complete computer program created using the chromosome along with the corresponding BNF grammar).

The mapping from genotype to phenotype will be a result of the set of production rules represented in the grammar. The phenotype is evaluated using a fitness function and simulator. Based on a fitness score and other attributes, individuals will then be selected to be "parents" in a new generation of individuals. This cycle ends when a suitable solution to the task has been evolved. Our work this summer entails the writing of a GE system to generate a suitable controller program for the Finch robot that enables the robot to navigate simple obstacle arrangements.

Figure 2: A Comparison of Grammatical Evolution and Natural Biology



III. Literature Review

The Bio-Computing and Developmental Systems (BDS) Group [BDS2014], Suggest that there are several advantages to GE:

- Complete decoupling of the solution space (the grammar) from the evolutionary mechanism
- Simple representation allows the use of any search algorithm that can operate on binary strings
- Representation is highly efficient by representing high level structures as binary strings

Behavior of the system can be changed by modifying the grammar. The core GE system never has to change. This means that we can change the problems that the robot is trying to solve (and increase its abilities as our work progresses).

Our research adopts these notions and applies GE to the programming of autonomous robots. The basic concept of GE was initially demonstrated successfully using three different types of problems, namely, symbolic regression [Ryan 98a], finding trigonometric identities [Ryan 98b], and a symbolic integration [Ryan 98c]. Using a BNF definition to output a high level program by mapping a binary genotype onto a phenotype, results show [Ryan 98a] that BNF mapping techniques allowed for the system to be language independent, and, theoretically generated arbitrarily complex functions. We plan to build upon this base by applying the technique in the more utilitarian task of programming a robot.

Techniques of GE were successfully applied to creating organized computer code by evolving programs using functions. When compared to evolving code exclusively in the main body of a program, the use of evolved function definitions led to improvement in overall performance [Ryan 98b]. The system we are building will make use of these findings by evolving several function calls. The generated functions will belong to the Java-Finch robot application program interface (API).

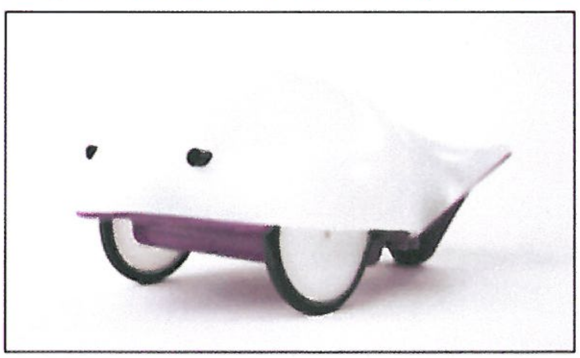
During a symbolic integration problem [Ryan 98c], it was found that a faulty generational selection mechanism produced some invalid results. These invalid chromosomes were passed onto the following generations preventing the evolution of a correct solution. As a remedy, a mechanism ensuring such individuals would be less likely to survive was put into place. This new selection mechanism caused a 60 percent enhancement in the performance of the system [Ryan 98c]. We expect to implement a similar preventative measure in our system. A good selector plays an important role in any GE system because it is the force which chooses which genomes will express their selves in future generations.

IV. Proposed Research and Expected Outcomes

The main objective of this research is to build a grammatical evolution system "Grammatical Evolution for the Finch" (GEF) that automatically generates a computer program that controls a Finch robot solving simple problems such as obstacle avoidance, object acquisition, and escaping from a room in the dark. That is, we are building a computer program to create another (robot control) program. In the future, the framework developed and the implementation deployed, will enable a robot to "self-program" by deploying GEF onto the robot.

The Finch robot was developed at Carnegie Mellon under NSF funding for computer science education to help teach programming skills. It is ideal for this research because the robot itself is not overly complex and is built to be directed to do simple tasks using simple programs. Because GEF will be *automatically* writing computer programs, simple programs are what we expect to be generated.

Figure 3: The Finch Robot developed at CMU

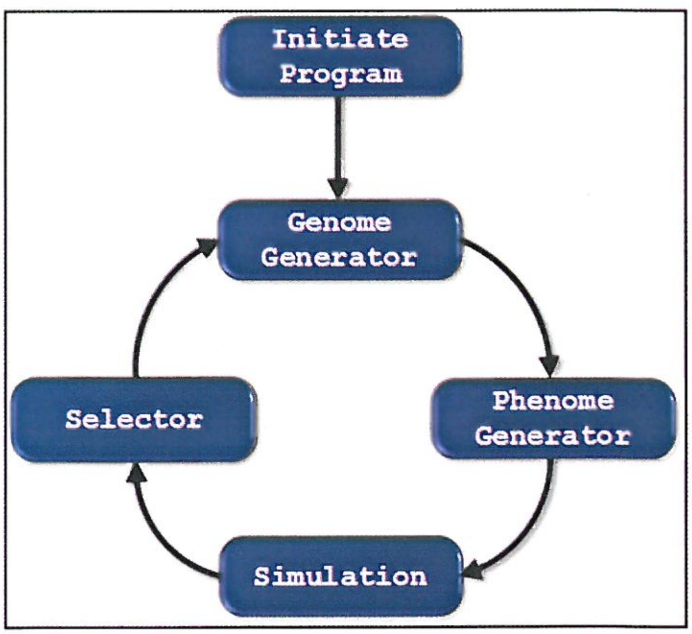


Flow of Program GEF

The GEF system is iterative and executes a cycle of steps for each generation of chromosomes, which runs until a reasonable program has been evolved. Starting with a genomic generator, GEF produces chromosomes consisting of a sequence of decimal numbers. Each chromosome or "individual" will derive a phenotype (which is a complete computer program created using the chromosome along with the corresponding Java-Finch BNF grammar).

The conversion from genotype to phenotype will be a result of the set of production rules represented in the grammar. The phenotype is evaluated using a fitness function and simulator. Based on a fitness score and other attributes, individuals will then be selected to be "parents" to continue in a new generation of individuals. This process continues in GEF until a suitable robot controlling program has been created. The program that GEF will evolve will direct a Finch robot to navigate around simple obstacles in a room. Figure 4 illustrates this algorithm.

Figure 4: Framework Diagram for GEF System



GEF Program Construction Plan

Building GEF will be a five part construction process that will use the techniques of grammatical evolution to create a computer program that controls a Finch robot. The process is described here.

1) Define an operational framework of the five system modules so that each module can be refined, improved, and replaced independently as our work progresses
2) Implement the modules in the Java programming language
3) Define a subset of the Java programming language that can be implemented in GEF and possesses the richness of capability to enable our robot to solve the problem at hand.
4) Test the derived programs generated by GEF on a simulator (that the PI is currently developing)
5) Deploy the very best of the generated programs onto the FINCH robots and testing them in a an appropriate environment

Work Accomplished To Date

The PI has been building robots for several years as a hobby. He has previously used and manually programmed Finch robots. In the past two months, the PI has started on building GEF by writing the following pieces that play a role in the components shown in

Figure 4.

1. the Finch Simulator (used to compute the Fitness Function)

2. the Genome Generator

The PI has also started to encode the JavaFinch programming language into a BNF representation.

Remaining Work

Again using Figure 4, the remaining work to be done in order to start running and evolving code consists of:

1. writing the Phenome Generator
2. writing the Selector
3. writing the Initiate Program
4. writing the Main Driver program that connects all system components together

V. Timeline

1) March – May 2011

   a. Continue design of overall system

   b. Finish encoding of the JavaFinch language into BNF

2) Summer 2014

   a. Week 1: June 23 -- June 29

      - Encode the BNF into Java

   b. Week 2 – 4: June 30 – July 18

      - Write the Phenome Generator, Selector, Initiate Program, and Main Driver

   c. Week 5 –6 July 21 – August 1

      - Run GEF 24-7 to evolve programs
      - Deploy and Test acceptable programs on the Finch

   d. Week 7 – 8 August 4 – August 15

      - Perform analysis and data interpretation

3) Fall 2014

   a. Write the final research report

   b. Write and present a conference paper

VI. Budget

Charles Peabody stipend	$3,000
Dr. Seitzer stipend	$1,800
Finch Robot	$100
Conference travel expected costs	$1700

Student registration $200

Student travel and meals $500

Faculty registration $500

Faculty travel and meals $500

Total (with Conference funding) $6600

Total (without funding of Conference Expenses) $5000

An alternative way to fund the conference presentation is through the faculty advisor's awarded Individual Development Grant recently awarded on 2/25/2014.

VII. Conclusion

In summary, during the funded period of research, from June 23, 2014 through Aug 15, 2014, the researchers plan to create a GE system that will create programs to control the Finch robot so that it can perform simple tasks. We ask for funding for student and faculty stipends ($4800), one Finch Robot ($100), and possible funding for expected conference expenses ($1700).

Respectfully Submitted,

Charles Peabody

VIII. References

[BDS 2014] "Grammatical Evolution: Evolution of Computer Programs Through grammars." The Biocomputing and Developmental Systems (BDS) Group. n.d. Web. 8 Feb. 2014. <http://bds.ul.ie/research.html>.

[Booker 1989] Booker, L.B., Goldberg, D.E., Holland, J.H. Classifier Systems and Genetic Algorithms. Artificial Intelligence, 1-3 (40), 235-282, September 1989.

[CMU 2014]*Finch API - Java*. Program documentation. *Finch API*. BirdBrain Technologies / Carnegie Mellon University, n.d. Web. 22 Feb. 2014.

[Russel 2010] Russell, Stuart J., Peter Norvig, and Ernest Davis. "Intelligent Agents." Artificial intelligence: a modern approach. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[Ryan-Col 1998] Ryan C., Collins J.J., O'Neill M. Grammatical Evolution: Evolving Programs for an Arbitrary Language. Lecture Notes in Computer Science 1391. First European Workshop on Genetic Programming 1998.

[Ryan-On 1998] Ryan C., O'Neill M., Collins J.J. Grammatical Evolution: Solving Trigonometric Identities. In proceedings of Mendel 1998: 4th International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets. June 24-26 1998, Brno, Czech Republic, pages 111-119.

[Ryan-On2 1998] Ryan C., O'Neill M. Grammatical Evolution: A Steady State approach. In Late Breaking Papers at the Genetic Programming 1998 Conference, University of Wisconsin, July 22-25, 1998. Madison, WI: Omni Press.

[Warford 2010] Warford, J. Stanley. Computer systems. 4th ed. Sudbury, Mass.: Jones and Bartlett Publishers, 2010.

Charles Peabody: Personal Statement

My name is Charles Peabody. I am a Combat Veteran who, after serving 8 years in the United States Marine Corps, is taking my career in a new direction. Since my Honorable Discharge from the Marine Corps, I earned an Associate's degree in Mathematics and Science at Community College of Allegheny County in Pennsylvania. Currently an A&S junior at Rollins College, I am seeking my bachelor's degree in Computer Science.

My desire to participate in the Student-Faculty Collaborative Scholarship Program (SFCS) stems from my intention to grow both professionally and intellectually. I eventually want to go to graduate school and therefore, looking forward to my involvement with the SFSC program, I see the opportunity to learn and to be involved in research. The proposed research will benefit me in that it will strengthen my skills as a computer scientist. With a greater understanding of some of the field's complexities, I feel I am provided a chance to excel and establish myself a more competitive candidate for future endeavors.

Student Contract - 2014

As a participant in the 2014 Student-Faculty Collaborative Scholarship Program I, **Charles Peabody**, will contribute intellectually and physically to my chosen scholarly work to the greatest extent possible. I agree to be present for work at the times and places specified by my faculty mentor, **Dr. Jennifer Seitzer**, and I understand that I will be paid **$3,000** as a stipend for working between **June 23, 2014** and **Aug 15, 2014**. I understand that the purpose of the Student-Faculty Collaborative Scholarship Program is to allow me to participate in original scholarship, and that the success of the program depends upon my dedication to it. With this in mind, under the guidance of my faculty mentor I will accomplish the following goals prior to the final date listed above.

1. Daily meetings Monday through Friday unless otherwise agreed upon (e.g., for contingent situations such as out-of-town meetings, etc.)
2. Attendance at special talks and events that would help the Student researcher in his studies as directed by Dr. Seitzer
3. The completion of development of the parts of the GEF software system as assigned
4. The software development of the Finch Robot Simulator to be used for Fitness function evaluation in the GEF system
5. The testing of the GEF system and deployment of select controller programs on the Finch robot
6. The composition of the final report documenting outcomes and results (to be completed Fall 2014)

I understand that while participating in the program I will be subject to the student rules of conduct and those infractions of these rules will be dealt with by the existing student judicial process. I further understand that violation of the College's non-toleration policy for drug use and paraphernalia will result in immediate expulsion from the program.

Charles Peabody, PI — Feb 26, 2014

Dr. Jennifer Seitzer — Feb 26, 2014