

**"Population Based Incremental Schema Learning"**

Matt Dominski (Computer Science)

Advised by Dale Brown (CS)

**Abstract**

Evolutionary algorithms are used a lot to solve non-polynomial problems. This works especially well since it has already been shown that they are able to find near-optimal solutions. This project will focus on the genetic algorithm (GA) and the population based incremental learning algorithm (PBIL). This paper will then take the PBIL and add functionality to the PBIL to create the population based incremental schema learning (PBISL) algorithm which uses the notion of schemata from the GA. The objective of this paper is to create a PBISL and compare it against a PBIL and a genetic algorithm. This comparison will be done by comparing the results of different problems like the parity, 0/1 knapsack and the traveling salesman problem.

\*\*\*\*\*

**"Mobile Robotics: The study of the Mindstorms Lego NXT Robotics Kit and its Applications in Autonomous Robotics"**

Richard Lawrence (Computer Science)

Advised by Denise Byrnes (CS)

**Abstract**

Mobile robotics refers to the construction of a mobile mechanical device that has some level of human intelligence and is capable of roaming freely without interference. In such a device, both hardware and software components are integrated to give the device the ability to sense and manipulate its environment.

This project seeks to analyze the Mindstorms Lego NXT robotics kit and its applications in mobile robotics. The NXT robotics kit is a relatively inexpensive robot that supports numerous creations and inventions within the bounds of its programmability and sensing capabilities. Though there are limitations to its capabilities, the NXT does offer a great deal of sophistication to users in mobile computing problem solving and is more than adequate for this project. In this project each individual sensor is tested to demonstrate its appropriate function, range of values and suitability for combined sensor input to analyze and make decisions based on specific testing environments. An example of this would be a simple touch sensor, whose range of values are the real numbers 0 and 1. At rest the sensor relays a 0 and when pressed it returns a 1. This sensor could then be used as a bumper in the implementation of an obstacle avoidance robotic car. A simple algorithm could be designed to tell the car to reverse and make a 90 degree turn if the bumper is pushed, then move forward again. Careful consideration and assessment is given to the use and implementation of all algorithms in programming solutions.

\*\*\*\*\*

**"Improving The Artificial Intelligence Component Of Warcraft II With Genetic Algorithms And Dynamic Scripting With Conditional Weights"**

Zachary Patterson (Computer Science)

Advised by Dale Brown (CS)

## **Abstract**

Everyday a new game comes out with better graphics than the ones before it. They have huge interactive worlds and deep and involving plot lines. But they use the same artificial intelligence techniques that the games a decade before used. It is the goal of this project to apply the techniques of academic AI to the realm of game AI. This project contains two parts. The first part trains a genetic algorithm to optimize the Grunt Rush strategy. The second part attempts to improve the dynamic scripting algorithm. It keeps track of rules that were successful together and encourages them to be paired together again. Some of the genetic algorithms solutions were able to beat the default script consistently. Dynamic scripting was not able to consistently beat the default script even with the conditional weights improvement.

\*\*\*\*\*

## **"Donald Davidson and Natural Language Processing"**

Joe Pletcher (Computer Science & Philosophy)

Advised by Dale Brown (CS) and G. Thompson (Philosophy)

## **Abstract**

This project aims to first explain and defend Donald Davidson's philosophy of language, against general objections as well as specific objections to computational application of Davidson's T-theories. The problems associated with applying a T-theory to a computer are investigated, along with the minimum requirements a computer would need to be able to apply a T-Theory. This project shows that while a computer might not currently possess the prerequisites for computational application of a T-theory, that there is nothing fundamentally or inherently in a computer's structure that would prevent such application. In addition, this project shows that a T-theory is sufficient for language understanding, and thus that by proving computational understanding of a T-theory one is proving computational language understanding.

The second section revolves around syntactical rules. Davidson requires sentences to be well formed, and his syntactical rules are a part of his T-theories, yet they must be separate from semantics to avoid regression. This opens up an avenue for computers: can a computer automatically generate syntactical rules given a sample text. Using a partially statistical, partially rule based approach the project shows how using distributional similarity and substitutability, and part of speech tags allows one to build a parser for random sentences off of a trained corpus.

\*\*\*\*\*

## **"Leading the Virtual Orchestra"**

Matthew Snively (Computer Science)

Advised by Denise Byrnes (CS)

## **Abstract**

With a number of technologies both new and old at hand, it is now possible to develop a program that can realistically mimic conducting an ensemble. The Wii Remote, acting as a baton, can send motion sensor data to a computer using Bluetooth wireless technology. The computer can then use the data to manipulate a MIDI file as it is being played in real time to match the actions of the conductor. Such an achievement was not previously possible without using expensive, custom-designed equipment and a powerful computer.

\*\*\*\*\*

## **"Making Monads: Binding to a Specific Type of Computation with a Return to Familiar Themes"**

Benjamin Strecker (Computer Science and Mathematics)

Advised by Simon Gray (CS) and John Ramsay (Math)

## **Abstract**

In the world of programming, there are many kinds of languages from which to choose. These varied languages each provide some easy solutions to some problems, but there may be other problems that the language inhibits. This thesis examines functional programming, beginning with the lambda calculus and moving up to the functional language Haskell. A mathematical object called a monad will be introduced, and its use will be demonstrated to solve problems that are normally difficult in many functional languages. One specific problem is parsing. A parser for a subset of the toy language Triangle will be constructed, followed by a contextual analyzer for the same language. Both of these programs will demonstrate how to effectively use monads in applicable situations.

\*\*\*\*\*

## **"Implementing Optimizations in the Joeq Virtual Machine"**

David Sullivan (Computer Science)

Advised by Simon Gray (CS)

## **Abstract**

Virtual machines are becoming more and more prominent in both the business and consumer worlds. They provide flexibility and cost-efficiency, however they can sometimes have a serious drawback: performance. The main focus here is the implementation and comparison of various optimization strategies for high level languages, particularly Java, using Joeq as a testing platform. These techniques will be built into Joeq, then sample Java programs will be run, and the performance measured. This paper also examines some other virtual machines and how they work, current virtual machines on the market today, and current optimization techniques, and attempts to use those to highlight new ways to optimize virtual machines today. Dispelling the notion that virtual machines always carry a significant performance price is another goal. Some of the optimization strategies and virtual machines discussed actually give better performance than natively compiled code. Transmeta's Crusoe, HP's Dynamo, the Java Virtual Machine, Jikes and Joeq will be examined, as will interpreters, binary translators, dynamic binary optimization, and related optimization strategies such as profiling, predecoding, direct and indirect threading, code reordering, translation blocks, and staged emulation.

\*\*\*\*\*

## **"Simulating a Charged Spherical Pendulum in Time-Varying Electric and Magnetic Fields"**

Mark Wellons (Computer Science & Physics)

Advised by Denise Byrnes (CS) and Todd McAlpine (Physics)

## **Abstract**

For this independent study we (Mark Wellons, Frank W. King, and Todd C. McAlpine) constructed a simulation of a damped, charged, spherical pendulum in time-varying electric and magnetic fields and studied the pendulum's long term behavior. By analyzing a large number of oscillating electric field amplitudes and a number of constant magnetic field strengths, sets of system parameters that force aperiodic behavior in the pendulum were mapped. We find that as the electric field amplitude increases, the pendulum's long term behavior moves through regions of periodicity and aperiodicity. In addition, the qualitative effect of the magnetic field on the pendulum's periodicity is demonstrated. We find that the magnetic field narrows the windows of aperiodicity, and this effect increases as the magnetic field strength increases. Each set of system parameters is independent of the others, so computation is parallelized using an Xgrid cluster. Further analysis of the pendulum's aperiodic behavior could rigorously show the system to be chaotic.