

# The Search for a Universal Turing Machine

Steven Dee, Justin Gray, Josh Lee, Neil Sandburg

March 26, 2009

# Overview

## 1 Team Members and Roles

## 2 Overall Objectives

## 3 Progress Report

- Schedule
- Progress
- Risk Assessment

# Team Members and Roles

- **Steve Dee:** Database managment, fitness evaluation algorithm
- **Justin Gray:** Genetic Algorithm implementation
- **Josh Lee:** Turing Machine modeling
- **Neil Sandburg:** Web visualization

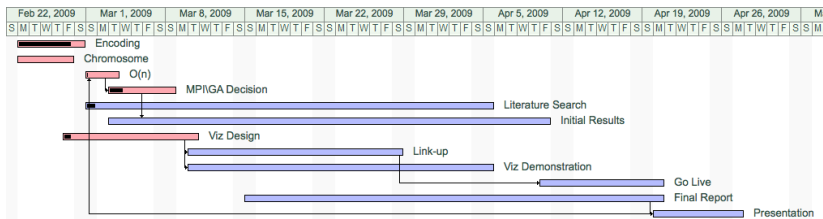
# Overall Objectives

Can we use the discovery of Universal Turing Machines as an analogy for the evolution of complex organisms?

- Test the possibility of discovering a set of Universal Turing Machines via a Genetic Algorithm optimization
  - Short optimization with a limited run time
  - Long optimization with a much longer run time
- Create a web application to publish the results of the optimization
  - Real-time link to database being populated by optimization results
  - Allow user submission of turing machine to test if it exists in the current population



# Schedule



- **Encoding**: Definition of the encoding method for Turing Machines completed
- **Chromosome**: Implementation of the chromosome class for the AI4R Genetic Algorithm package completed.
- **Initial Results**: Initial Turing Machine Optimization run completed. Data analyzed to ensure the viability of optimization scheme
- **Go Live**: Ruby on Rails visualization application goes live with link to genetic algorithm running perpetually.

# Progress

## Genetic Algorithm Optimization:

- Turing Machine implementation
- Genetic Encoding: a set of 5 tuples, representing the state transition table of a turing machine
- Selection of a Genetic Algorithm: AI4R (<http://ai4r.rubyforge.org/>)
- Chromosomal implementation begun
  - initialization function complete
  - fitness scoring algorithm established: score based on adaptability

## Web Visualization App:

- Population database created
- Conceptual design begun

# Risk Assessment

- Nothing interesting happens: No interesting TM's result from any of the optimizations. We don't have time to develop a new fitness scoring algorithm and test it.
- Something interesting happens, but we don't recognize it: Our literature search turns up UTM's, but none of them show up in optimization. TM's that do turn up appear to be very "adaptable", so they might be UTM's.
- Time constraints: may not have enough time to implement the web visualization tool.