Project Proposal

Riyansh Karani (UIN - 672093027, Net ID - rkaran3)

Consider an imperative programming language wherein the state of the program is described by the state of a regular 8x8 chess board. All the "valid moves" in a game of chess, that lead to a transition of the board from one state to another are valid state transitions in the programming language. A ***program*** in the language is a pair consisting of

1. an ***initial state*** that consists of the state of the board along an identifier that signifies whether the current state represents a check mate.

2. an ***ordered list of valid instructions***, wherein the each valid instruction is a valid move represented as a string as per the algebraic notation. For example, the instruction Be5 represents the move wherein bishop moves to the grid e5. The program "halts" after all the instructions have been executed. Note that in this definition of halting, it is implicit that that no instruction can follow an instruction the leads to "checkmate", since the instructions that follow would not represent valid moves in the game of chess.

Given this definition of a program in this programming language, there are two different kinds of tasks that I that I propose to work on as a part of the project:

Concrete tasks

- Formalize the abstract syntax of this programming language.
- Formalize the small step and big step semantics of this language that describes the behavior of the program.
- Implement an interpreter in OCaml that correctly executes a program.

Abstract (open ended) tasks

- Try and come up with proofs for interesting known properties of the game of chess. For example, it is true that for any endgame in the game of chess wherein the board contains two kings and a queen, it is possible for the game to end with a checkmate.
- See if it is possible to "discover" properties about the game of chess by following the operational semantics of the language.
- Try to come up with an interpretation (a level 2 semantics, for the lack of a better phrase) that represents computations in the traditional sense. For example, is it possible to represent a function that computes the 5th Fibonacci number in this programming language.