NOTE: THIS IS A MARKDOWN FILE, BEST VIEWED IN THE INCLUDED PDF, OR A MARKDOWN VIEWER.

Project 3: Prolog

By: Kevin Bohinski & David Vassallo

11/18/2016

CSC 435 Programming Languages

Readme File

Files

Within the zip there should be three files:

```
\-p3.zip
    | kb.pl
    | readme.md
    | readme.pdf
```

`kb.pl` contains our source, and `readme.md` or `readme.pdf` contains this readme.

Running the Project

- Unzip the folder.
- Load kb.pl into your prolog interpreter of choice.
- Type `solve.` into your interpreter.
- Done! Output should be printed.

Input & Output

Our file requires you to query `solve.` to begin running through the logic, there is no main function that is ran on launch. Attempts to solve riddle but encounters runtime error after certain point.

Positives

- Project has all required rules implemented.
- Has an incomplete breadth first search implemented to search for number of possible solutions.

Limitations and Bugs

- Project encounters runtime error after certain point.

Conventions

- Attempted to follow conventions found in the prolog documentation.

Abstractions

Abstractions are not a part of the logic programming paradigm.

Data Structures

We made extensive use of lists, rules and atoms.

Lists

Lists were used to model states and maintain a history of states.

Rules

Rules are the basis of logic programming and provides the logical constraints for the interpreter to perform the task.

Atoms

Atoms are used throughout, one example of the use may be to determine if it is safe for the hens to travel.

Overview of Rules

We implemented the rules for working through the solution. They are as follows:

Crossings:

The `cross` rules represent the eight possible ways to move across the river.

Control Flow Rules:

Overarching functions to manipulate the data and solve the program.