Prolog Programming

11/27/2020



By Connor Bartal & Sean Rogers

**What is Prolog:**

Prolog stands for “Programming in logic. A person typically usually a programming language to tell a computer to perform whatever task that they desire. In other languages the programmer tells that the computer what a variable is. For example, you can define int a = 3 (telling the computer the variable a is the integer 3). The program than can set evaluate or change that variable based on the program. However, in prolog the computer is told less how to complete the task. In prolog a set of facts are defined, and these facts could be used to define characteristics, such as (“Smith is the father of Adam”). The Prolog program can than ask the computer about the facts already given and the computer well return the provided answers.

Diagram, shape, square

Description automatically generated**Prolog Graphs**

The most interesting thing we found is that prolog can easily represent graphs with

its rule system. For example, the graph on the right could be represented with the following

code.

edge(1,2).

edge(1,4).

edge(1,3).

edge(2,3).

edge(2,5).

edge(3,4).

edge(3,5).

edge(4,5).

connected(X,Y) :- edge(X,Y) ; edge(Y,X).

With the use of this disjunction, (;) It is the same as writing the following code. Which connects the graph both ways, this can of course be changed to connect one way or to include lengths to find the shortest distance.

connected(X,Y) :- edge(X,Y).

connected(X,Y) :- edge(Y,X).

This allows a connection between all the edges allowing for powerful algorithms to find the paths, graphing, or even mathematical formulas. Prolog sets up a powerful and easy system to build graphs, with nodes, edges, lengths because of its easy to use rule system of predefined facts. Graphs are a powerful mathematical tool in some cases and prolog just makes it easier to use.

**Ease of Use and Understanding**

We decided that we like prologs easy to read fact system and found it interesting that you could write English statements out than define it in prolog facts. For example, “John is the father of Jason” could be written in prolog in the following form

father\_child(John, Jason)

parent\_child(X, Y) :- father\_child(X, Y).

Even those with very little programming experience could look at that fact and understand the English statement that was made earlier. We believe that the system could even be used to teach new programmers, programming logic as it is easier to read and more compact than other languages.