CS4402 Learning Jounal 8

* Describe what you did. This does not mean that you copy and paste from what you have posted or the assignments you have prepared. You need to describe what you did and how you did it.
* Describe your reactions to what you did
* Describe any feedback you received or any specific interactions you had. Discuss how they were helpful
* Describe your feelings and attitudes
* Describe what you learned

Another set of questions to consider in your learning journal statement include:

* What surprised me or caused me to wonder?
* What happened that felt particularly challenging? Why was it challenging to me?
* What skills and knowledge do I recognize that I am gaining?
* What am I realizing about myself as a learner?
* In what ways am I able to apply the ideas and concepts gained to my own experience?

During this week of learning, I gained some exposure to prolog.

I started with the reading assignment and learned some basic concepts of logic programming.

The basic concept of logic programming is declarative and consists of only mathematical logic. Each set of formulas for the same predicate is interpreted as a procedure. A particular computation is defined by submitting a goal. The prolog compiler is an inference engine that searches a possible proof of the goal from the program.

Below are some prolog features.

Facts and Rules: Prolog programs are built on facts and rules. Facts are like pieces of information, and rules define relationships or conditions.

Predicates: Predicates are used to define facts and rules. They help you make statements about your program.

Queries: You can ask Prolog questions by making queries. The system will try to find a solution based on your facts and rules.

Unification: This is a critical concept in Prolog. It's the process of finding values that satisfy a query by matching variables in the query with terms in your facts and rules.

Backtracking: If Prolog can't find a solution, it backtracks to previous choices and explores alternative paths. It's like exploring a maze to find the exit.

Recursion: Prolog is well-suited for recursive programming. It can define rules that refer to themselves, making it great for solving problems that involve repetitive patterns.

Variables: Variables are used to represent unknown values. They can be unified with specific values during the execution of your program.

Cut Operator (!): The cut operator is used to control backtracking. It prunes alternative choices, helping to make the program more efficient.

Lists: Prolog has built-in support for lists. You can manipulate and work with lists using various predicates.

Databases: Prolog can be used to build knowledge databases, making it suitable for applications like expert systems and artificial intelligence.

The main differences between functional programming and logic programming are below:

Logic programming uses two-way unification which is a stronger pattern matching used in functional programming.

Functional programming is one-directional in that given arguments the program returns.

Facts example father(albert,Jeffery). The formula follows functor, constant, and period at the end.

Rules are a relationship. Parent(x,y) :- father(x,y) being father implies that the father is a parent.

All the facts / atom starts with a lower lowercase letter. The questions start with ‘?’ is questions.

Father => parent. A combination of rules can establish a new relationship.

X y , y z = > grant parent X Y.

The question is separated by a comma and it’s equivalent to ‘and’.

There are four types of terms: atoms, numbers, variables, and compound terms.

Atoms are string that starts with lower case, upper case letters number, and underscore.

It must start with lower lowercase letter. A variable must start with a Capital letter or underscore.

Sibling relationship x\=Y.

Logical equality and unifynity is different. I gained an understanding of how prolog walks from the logic tree by recursion.

A program is a database of clauses that are assumed to be true.

Overall, I felt more confident with different types of programming languages. This course broadens my view of different types of programming languages and how to handle computational problems with them.