Prolog

* **Predicates** – Relations
* **Horn Clauses** – Specify Relations using the form P if Q1 ^ Q2 ^ … ^ Qk
* **Rule** – Relation with k>0 conditions
* **Fact** – Relation with k=0 conditions
* **Terms** – Facts, Rules, Queries, and even data
* **Functor** – Atom of a compound term
* **Arguments** – subterms of a compound term
* **Backtracking** – Prolog process where if a term fails, it goes back and retries previous terms and if there aren’t any previous terms then it tries the next clause
* **Unification** – Prolog process to match two terms
* **Instance** – An instance of a term is a less general version of the base term most likely with some of the arguments specified to be something
* **Open List** – List that ends with a variable ex. X = [1, 2, 3, 4 | Y]
* **Cut** – If exclusive rules aren’t an option, one can use cuts to prevent backtracking where a red cut alters the program’s behavior while a green cut is used to increase efficiency

Summaries

* + Logic programming deals with relations, which are more powerful than functions.
    - Traditional arguments and results are treated uniformly.
  + Horn clauses can be used to specify facts and rules, using universal and existential quantifiers.
  + Prolog is a logic programming language.
    - Everything in a Prolog program is specified using terms.
    - It approximates pure logic, as clause order matters.
    - Backtracking is used to find all possible solutions.
  + A “no” response merely indicates that Prolog cannot prove our query. It does not mean it is false.
  + Prolog uses unification to answer our queries.
    - All variables in unification do not have to be bound.
    - All Prolog essentially does is try to unify terms and backtrack if it cannot.
  + Expressions are evaluated using the is operator.
    - All variables to be evaluated must be bound.
  + Exclusive clauses should be written as such.
  + Prolog has only one data type: term.
  + Everything is represented as terms, including clauses and queries.
  + Higher order predicates require other predicates as arguments.
  + Terms can be visualized as trees: the parent represents the functor, and the children represent the arguments.
  + Backtracking in Prolog is a powerful feature that naturally generates all solutions to a problem.
    - In particular, clauses are not exclusive unless explicitly written as such.
  + A cut is a mechanism to limit backtracking.
    - Green cuts are used for efficiency.
    - Red cuts are used to alter the program’s behavior.
  + A general-purpose relation such as not or \+ should be used wherever possible for clarity.
  + The λ-calculus is a functional model of computation based on function abstraction and application.
  + The only two operations are !-conversion and "- reduction.
  + The pure calculus does not include numbers, Booleans, or types.
  + However, we can define these in the calculus itself.
    - Church numerals and Booleans are commonly used.
    - With these, we can define arithmetic and logical operations.

Lambda Calculus













