Prolog program

* Made up of sequence of clauses
  + Facts or rules

Logical deduction

* Individual components of clauses are called literals
* Clause usually made up of clause A1 or … An if B1 and … Bm
* In prolog we always have clauses with n = 1 and are called Horn clauses
* 1st literal in the clause is the head of the clause
* Any remaining literals (ex any after the ( :- ) form the body of a clause
* A fact is a horn clause with an empty body
* Any other clause is a rule

Resolution

* Prolog uses an inference rule called resolution
* A query ?- a1, a2, … , an may be resolved with a fact a1 to yield a new query ?- a2, a3, … , an
* A query ?- a1, a2, a3, … , an may be resolved with a rule a1 :- b1, b2 , … , bm to give the new query ?- b1, b2, … , bm, a2, a3, … , an
* A query ?- a may be resolved with a fact a. This gives the empty clause true.

Resolution strategies

* Chooses the first literal in the current goal and resolve the current goal with the earliest clause in the program whose head matches the chosen literal
* If none of the program clauses match the goal, backtrack to the previous goal and look for another clause to resolve with that
  + If a decision leads down a path, eventually returning false
  + Then go back to the most recent decision point and make the next decision
* Draw a resolution tree to show the backtracking history
  + Helps to debug
* Because of these resolution strategies the order of the literals in a clause and the order of the clauses within a program matter
  + Means a lot of mathematical logic doesn’t apply

A picture containing table

Description automatically generated