Find connections in this social network
Logic Programming

✦ Say what you want, not how you want it done

✦ A program written in a logic programming language consists of:

• A set of **facts** about objects and their relationships

• A set of **rules** about objects and their relationships

• A set of **queries** about objects and their relationships
Horn Clauses

- A Horn clause is a statement that connects a single predicate to a set of conditions for that predicate to apply.

  - connection(X, Y) :- friend(X, Z), friend(Z, Y).

  - The way to read the clause is that the connection predicate applies to X and Y if X and Z are friends, and Z and Y are friends.

  - The predicate on the left applies to the variable X and Y only if all of the predicates on the right also apply to X, Y, Z.
Resolution

If $h$ is the head (the left side) of a Horn clause

isMammal(x) :- cat(x)

and it matches one of the terms (the right side) of another Horn clause

isAnimal(x) :- isMammal(x)

then that term can be replaced by $h$'s terms to form

isAnimal(x) :- cat(x)
Unification

- A simple predicate of the form $\text{cat}(x)$ is the head of a Horn clause with no terms, which means it is a fact.

- Binding a variable with a value is called instantiation.

- The process of identifying the set of values for a set of variables in a Horn clause that make it true is called unification.
Prolog

✶ Fact

- Consist of a particular item or a relation between items
- Should begin with a lowercase letter and end with a full stop.
- Can consist of any letter or number combination, as well as the underscore _ character.
- Should avoid the characters +,-,*,/, or other mathematical operators.
- Relation: more complicate facts
  - relation(<argument1>,<argument2>,...,<argumentN>).
  - Relation names must start with a lowercase letter.
  - Can have N arguments, N >= 0.
- A variable is a string of uppercase letters, lowercase letters, digits and underscore characters that starts either with an uppercase letter or with an underscore.
 Rules: make conditional statements

• Each rule can have several variations to specify alternative ways of proving a particular thing.

• Prolog starts from the first rule/fact. If it does not apply, Prolog tries the second one. The query fails if we run out of rules/facts.

• Variables with the same name in a rule have the same instantiation for each solution to a particular query.

• Identical variable names in separate rules are independent.