Typology of programming languages

Logic Programming

Prolog

Marriage between natural language processing and automated theorem-proving.

People behind Prolog

Alain Colmerauer
University Aix-Marseille, France NLP

Philippe Roussel
University Aix-Marseille, France NLP

Robert Kowalski
University Edinburgh, UK
Theorem Proving

Prolog, 1972

- Prolog (PROgramming in LOGic), first and most important logic programming language.
- Developed in 1972 in Marseille
- Relational rather than functional programming language
- Competitor to LISP for AI programming in 80's
- Adopted by Japanese for Fifth Generation Computing Project.

Prolog

- Prolog is a Declarative or logical Language.
- Prolog takes only facts and rules to arrive at goals.
- The programmer doesn't tell it how to solve.
- For solving logic and decision problems, Prolog is ideal.
- Typical applications: AI, Database apps, proving theorems, symbolic evaluation (i.e. differentiation).

How does Prolog work?

- Prolog is based on ‘Horn Clauses’
- Horn Clauses are a subset of ‘Predicate Logic’
- Predicate logic is a way of simply defining how reasoning gets done in logic terms.
- Predicate Logic is a syntax for easily reading and writing Logical ideas.

How does Prolog work?

- To transform an English sentence to Predicate Logic, we remove unnecessary terms.
- This leaves only the relationship and the entities involved, known as arguments.
- Ex: A pie is good = good(pie)
- The relation is ‘good’, the relation’s argument is ‘pie’.

Prolog basics

- **Term**: objects/data of the program
  - variables: unknown object
  - elementary: int, string, identifiers
  - compound: structured objects
- **Atom**: relation between terms
- **Clause**:
  - facts: relations that are known to be true by the programmer
  - rules: Used to infer other facts
- **Goals**: Part of program where queries are made

Prolog first example

```
likes(mary,food).
likes(mary,wine).
likes(john,wine).
```

```
?- likes(john,food).
false.

?- likes(john,wine).
true.
```

Prolog second example

```
male(alan).
male(gary).
female(margaret).
parent(alan, gary).
parent(alan, margaret).
mother(X,Y) :- parent(X,Y),
               female(Y).
father(X,Y) :- parent(X,Y),
               male(Y).
```

```
?- mother(alan,Mom).
Mom = margaret.
?- father(alan,Dad).
Dad = gary
```

Summary