

## Introduction to Prolog Programming

### Prolog

- Prolog stands for **P**rogramming in **L**ogic
- Although there are other Logic Programming languages, by far the most widely used is Prolog.
- All programming languages have both declarative (definitional) and imperative (computational) components
- Prolog is referred to as a declarative language because all program statements are definitional
- A Prolog program consists of *facts* and *rules* which serve to define relations (in the mathematical sense) on sets of values

2

Introduction to Prolog

26/01/2020

### Prolog...

- The imperative component of Prolog is its execution engine based on unification and resolution, a mechanism for recursively extracting sets of data values implicit in the facts and rules of a program
- Note that in logic programming:
  - Programs are written in the language of some logic
  - Execution of a logic program is a theorem proving process; that is, computation is done by logic inferences

3

Introduction to Prolog

26/01/2020

### Prolog...

- There are several implementations of PROLOG
  - SWI PROLOG - <http://www.swi-prolog.org/>
  - Turbo PROLOG - <http://www.fraber.de/university/prolog/tprolog.html>
  - Micro PROLOG - [http://www.lpa.co.uk/dow\\_fre.htm](http://www.lpa.co.uk/dow_fre.htm)
  - Visual PROLOG - <http://www.visual-prolog.com/>
- They have some differences in syntax

4

Introduction to Prolog

26/01/2020

## Computation vs. Deduction

- To **compute** we start from a given expression and, according to a fixed set of rules (the program) generate a result
- To **deduce** we start from a conjecture and, according to a fixed set of rules (the axioms and inference rules), try to construct a proof of the conjecture
- So computation is mechanical and requires no ingenuity, while deduction is a creative process

5

Introduction to Prolog

26/01/2020

## Computation vs. Deduction...

- Computation can be seen as a limited form of deduction because it establishes theorems. For example,  $15 + 26 = 41$  is both the result of a computation, and a theorem of arithmetic
- Deduction can be considered a form of computation if we fix a strategy for proof search, removing the guesswork (and the possibility of employing ingenuity) from the deductive process
- Logic program computation proceeds by proof search according to a fixed strategy. By knowing what this strategy is, we can implement particular algorithms in logic, and execute the algorithms by proof search

6

Introduction to Prolog

26/01/2020

## Why Prolog is not as popular as C/Java

- Mistaken at first as some universal computer language
- Not yet as efficient as C
- Support to Prolog takes effort, resources; companies are not willing to pay for it
- Its value not recognized by industry

7

Introduction to Prolog

26/01/2020

## What is Logic?

- A logic is a language. It has syntax and semantics. More than a language, it has inference rules.
- **Syntax:** the rules about how to form formulas; this is usually the easy part of a logic
- **Semantics:** about the meaning carried by the formulas, mainly in terms of logical consequences
- **Inference rules:** describe correct ways to derive conclusions

8

Introduction to Prolog

26/01/2020

## Facts and Rules

- Everything in Prolog is defined in terms of two constructs: the **fact** and the **rule**
- A **fact** is a Prolog statement consisting simply of an identifier followed by an n-tuple of constants. The identifier is interpreted as the name of a (mathematical) relation and the fact states that the specified n-tuple is in the relation
- In Prolog a relation identifier is referred to as a **predicate**; when a tuple of values is in a relation we say the tuple satisfies the predicate

9

Introduction to Prolog

26/01/2020

## Facts and Rules...

- Like "Jim is a child."
  - In PROLOG we write "child(jim)."
- "Joe is father of Jim."
  - father(joe, jim)
- "Jill is mother of Jim"
  - mother(jill, jim).

10

Introduction to Prolog

26/01/2020

## Facts and Rules...

- A **rule** is a Prolog statement which gives conditions under which tuples satisfy a predicate
- In a sense, a fact is just a special case of a rule, where the condition is always satisfied (i.e., equivalent to "true")
- A graph is very interesting, but not of much use if there is no way to specify more complex properties other than edge

11

Introduction to Prolog

26/01/2020

## Facts and Rules...

- Like "A parent is either a father or mother."
  - parent(X,Y) :- father(X,Y); mother(X,Y).
    - ";" is the logical disjunction **OR**
- "Two persons are siblings if they have the same parents."
  - siblings(X,Y) :- parent(Parent,X), parent(Parent,Y).
    - "," is the logical conjunction **AND**
- Variables are in Uppercase. They need to start with a Uppercase letter or "\_". Like "Parent" or "\_parent".

12

Introduction to Prolog

26/01/2020

## Example Program

FACTS of the program:

- `child(jim).`
- `father(joe, jim).`
- `mother(jill, jim).`
- `child(jan).`
- `father(joe, jan).`
- `mother(jill, jan).`

13

Introduction to Prolog

26/01/2020

## Example Program...

After loading the file in PROLOG. We can ask questions from PROLOG like:

- Is joe the father of jim?
  - `?father(joe, jim).`
- Who is the father of jim?
  - `?father(X, jim).`
- Whose mother is jill?
  - `?mother(jill, X).`

14

Introduction to Prolog

26/01/2020

## Example Program...

We can add some rules to our program.

- `parent(X,Y) :- father(X,Y); mother(X,Y).`
- `siblings(X,Y) :- parent(Parent,X), parent(Parent,Y).`

And we can ask questions from it:

- “Who are the parents of jan?”
  - `?parent(X,jan).`
- “Who are siblings to each other?”
  - `?siblings(X,Y).`

15

Introduction to Prolog

26/01/2020

## Exercise

- Given information about fatherhood and motherhood, determine grandparent relationship
  - John is father of Lily
  - Kathy is mother of Lily
  - Lily is mother of Bill
  - Ken is father of Karen
- Who are grand parents of Bill?
- Who are grand parents of Karen?

16

Introduction to Prolog

26/01/2020

## Exercise...

- In Prolog, we write the following for the given facts:
  - father(john, lily).
  - mother(kathy, lily).
  - mother(lily, bill).
  - father(ken, karen).
- Words like father, mother are called *predicates*
- A statement like father(john,lily) is called *an atom*, stating a true fact

17

Introduction to Prolog

26/01/2020

## Exercise...

- Rules for the grandparent relationship:
  - grandparent(X,Z) :- parent(X,Y ), parent(Y,Z ).
  - parent(X,Y) :- father(X,Y ).
  - parent(X,Y) :- mother(X,Y ).
- These are called *conditional statements*
- Capital letters denote variables, meaning “for any.”
- *For example:* the first rule above reads: For any X,Y,Z, if X is a parent of Y, and Y is a parent of Z, then X is a grand parent of Z.

18

Introduction to Prolog

26/01/2020

## Exercise...

- The complete Prolog program is:
  - father(john, lily).
  - mother(kathy, lily).
  - mother(lily, bill).
  - father(ken, karen).
  - grandparent(X,Z) :- parent(X,Y ), parent(Y,Z ).
  - parent(X,Y) :- father(X,Y ).
  - parent(X,Y) :- mother(X,Y ).

19

Introduction to Prolog

26/01/2020

## Exercise...

- Provide a query to Prolog; e.g.
  - ?- grandparent(john,bill)
- Prolog uses your program to do reasoning and answer the query, in this case, the answer by Prolog is YES/TRUE.
- If you post your query as:
  - ?- grandparent(Q, karen)
- Meaning who are grand parent of karen, Prolog answers NO/FALSE

20

Introduction to Prolog

26/01/2020

### Exercise...

- You can ask Prolog the question: who are the grand parents of bill by posting a query
  - `?- bill) grandparent(Q,bill)`
- Prolog answers
  - `Q = john`
  - `Q = kathy`

21

Introduction to Prolog

26/01/2020

### Exercise...

- You can ask Prolog the question: who are the grand children of john by posting
  - `?- grandparent(john,W).`
- Prolog answers
  - `W = bill`

22

Introduction to Prolog

26/01/2020

### Clauses

- **Clauses:** Prolog programs consist only of a sequence of **clauses**. All clauses are either **facts** or **rules**.
- Clause can run over more than one line or there may be several on the same line
- A clause is terminated by a dot character, followed by at least one 'white space' character, e.g. a space or a carriage return.

23

Introduction to Prolog

26/01/2020

### Other Prolog Terminologies

`animal(X):-dog(X).`

- The `:-` character (colon and hyphen) can be read as 'if'.
- *X* is called a *variable*. In this context *X* represents any value, as long as it is the same value both times.
- The rule can be read in a natural way as *X is an animal if X is a dog (for any X).*

24

Introduction to Prolog

26/01/2020

### Other Prolog Terminologies...

- We say that a goal *succeeds* or *fails*, or alternatively that it is *satisfied* or *cannot be satisfied*.
- The term evaluating a goal is used to mean determining whether or not it is satisfied
- Equivalently, we can say that a goal evaluates to **true** (i.e. succeeds) or **false** (i.e. fails). This all fits in well with the everyday definition of a goal as 'something to be achieved'.

25

Introduction to Prolog

26/01/2020

### Other Prolog Terminologies...

- Note that sometimes a goal entered by the user can be interpreted as a command, e.g.
  - `?-halt.`            'Exit from the Prolog system.'
- At other times it can be regarded as a question, e.g.
  - `?-animal(fido).`    'Is fido an animal?'

26

Introduction to Prolog

26/01/2020

### Atoms

- Atoms are constants that do not have numerical values. There are three ways in which atoms can be written:
  - (a) Any sequence of one or more letters (upper or lower case), numerals and underscores, beginning with a lower case letter, e.g.
    - `john`
    - `today_is_Tuesday`
    - `fred_jones`
    - `a32_BCD`

27

Introduction to Prolog

26/01/2020

### Atoms...

- but not
  - `Today`
  - `today-is-Tuesday`
  - `32abc`

28

Introduction to Prolog

26/01/2020

### Atoms...

- (b) Any sequence of characters enclosed in single quotes, including spaces and upper case letters, e.g.
  - 'Today is Tuesday'
  - 'today-is-Tuesday'
  - '32abc'

29

Introduction to Prolog

26/01/2020

### Atoms...

- (c) Any sequence of one or more special characters from a list that includes the following + - \* / > < = & # @ :
- Examples
  - +++
  - >=
  - >
  - +--

30

Introduction to Prolog

26/01/2020

### Assignment 1 – Q1

- Given a set of facts of the form father (name1, name2), that is, name1 is the father of name2.
  - a) Define a predicate brother(X,Y) which holds iff X and Y are brothers
  - b) Define a predicate cousin(X,Y) which holds iff X and Y are cousins
  - c) Define a predicate grandson(X,Y) which holds iff X is a grandson of Y
  - d) Define a predicate descendent(X,Y) which holds iff X is a descendent of Y

31

Introduction to Prolog

26/01/2020

### Assignment 1 – Q2

- Type the following program into a file
 

```
/* Dating Agency Database */
person(bill,male).
person(george,male).
person(alfred,male).
person(carol,female).
person(margaret,female).
person(jane,female).
```

32

Introduction to Prolog

26/01/2020



## Assignment 1 – Q2...

- Extend the program with a rule that defines a predicate **couple** with two arguments, the first being the name of a man and the second the name of a woman. Load your revised program into Prolog and test it. Clearly show the queries used in the testing process.

33

Introduction to Prolog

26/01/2020

END

34

Introduction to Prolog

26/01/2020