LOGIC PROGRAMS

- **Logic Rule**
- **Variables and Terms**
- **Lists and Recursion**
- **Logic representation of numbers**

when program is simple and no choice is involved, no difference between logic and Prolog

Prolog is defined by the order in which goals are satisfied and backtracking

LOGIC RULE

A logic program is defined as a finite set of rules (clauses)

$A \leftarrow B_1, B_2, \ldots B_n$. In Prolog $\leftarrow$ is :-

where LHS is the head and RHS is the body of the rule containing goals B_i

Note that this is the general form for all rules, facts, queries and called a clause (actually Horn clause)

- With n = 0 i.e. no body we have a unit clause, which is just a fact.
- With no head, we have a conjunctive set of goals

Answering a query is equivalent to determining whether the goal is a logical consequence of the program, using deduction rules

Variables and terms

A **logic variable** stands for single but unspecified individual.
A query is existentially quantified (In predicate logic)
? parent(tom,X). Does there exist an X such that …..

A **term** is the single data structure in a logic program and compound terms are terms, which consist of functor and one or more arguments

f(t1,t2….tn) where functor has name f, arity n
and each argument can be a term

e.g. father(tom, john, michael) can be expressed father(tom, children(john, michael))

A term is called ground if no variables, otherwise non-ground

Lists and Recursion

List is a binary structure:

- first argument is an element, the head of the list
- second argument is recursively the rest or tail of the list
- written as [X | Y] where X is the head and Y is tail of the list
- empty list denoted by [ ]

% indicates comment

```
%list( Xs) ← Xs is a list.
list( [ ] ).
list( [X | Xs ]) ← list(Xs).
```

Convention is that if X is head of a list Xs (plural) denotes tail

Proof tree of ?-list ([a, b, c]):

list([a, b, c]) ----- list([b, c]) ----- list([c]) ----- list([ ])

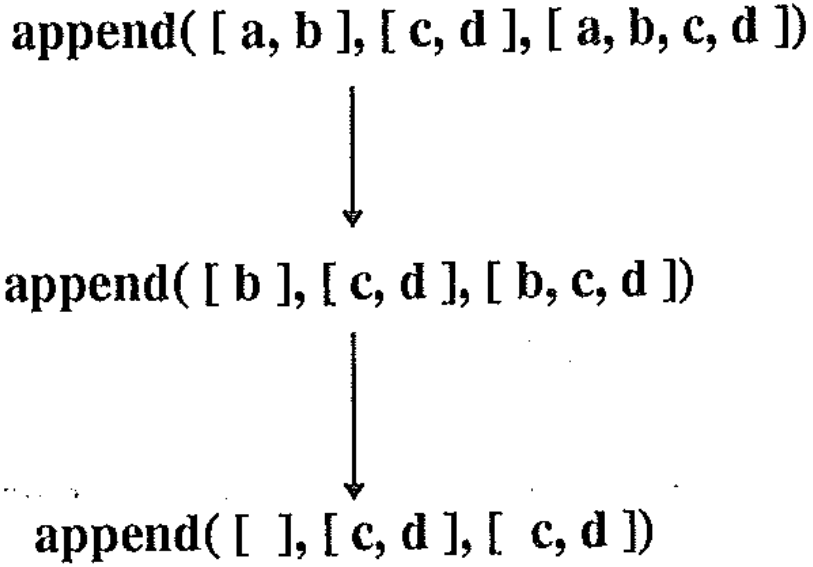
```
%member( Element, List) ← Element is an element of List
member( X, [ X | Xs] ).
member( X, [ Y | Ys] ) ← member( X, Ys).
```

- Declaratively, X is an element of a list if it is either the the head of the list or if it is a member of the tail of the list.

Example Append Query – no logic variable arguments

%append(Xs, Ys, XsYs) :- XsYs is the result of concatenating Xs and Ys
append([ ], Ys, Ys).
append([X | Xs], Ys, [X | Zs]) ← append(Xs, Ys, Zs).

Proof tree for ?- append([a, b], [c, d], [a, b, c, d]).



Arithmetic – logic representation not efficient

```
%type definition
%natural_number(X) ← X is a natural number
natural_number(0).
natural_number( s(X)) ← natural_number(X).
```

where s/1 **(successor term)** integer 3 rep. by s(s(s(0)))

this is logic representation of numbers, but too inefficient to be practical and Prolog uses built-in predicates