**Artificial Intelligence Workshop 3**

**Task 1-**  i\_play\_tennis(yes, X) :-

it\_is\_sunny(X), it\_is\_saturday(X).

it\_is\_sunny(Y) :- weather\_is\_fine(Y).

weather\_is\_fine(yes).

it\_is\_saturday(yes).

Query- i\_play\_tennis(X, yes).

**Task 2-**

1. The left hand part of a rule is called the rule head whilst the sub-goals on the right hand side are called the body of the rule.
2. Draw the solution tree for the bird(X) query whose execution is shown traced in section 2.2.

4) Put the following facts into Prolog's database: Rules 43 event(battle\_of\_hastings, 1066). event(plague\_of\_london, 1665). event(fire\_of\_london, 1666). event(man\_on\_the\_moon, 1969). Define a rule happened\_before(X,Y) which defines the relation that event X happened before event Y, and use it to obtain all such pairs of events. You will need to use the system predicate < 7). succeeds. Suitably instantiated variables can be used either side of < instead of numbers. Start the rule with 'happened\_before(X,Y) :-'.

Answer- happened\_before(Event1, Event2):- event(Event1, T1), event(Event2, T2), T1 < T2.

Query- happened\_before(Event1, Event2).

6) Introduce the following car facts into the database alongside those we have met previously: car(ford, capri, injection,uk, 2800, sports, 11200). car(alfa\_romeo, sprint, veloce, italy, 2000, coupe, 12500). car(volvo, 928, gls, sweden, 1400, hatchback, 6290). car(mitsubishi, colt, glx, japan, 1800, estate, 7420). car(mercedes, roadster, 280, germany, 2800, convertible, 18450).

Write a set of facts that gives points for the type of car body according to the following table: type points estate 1 saloon 2 hatchback 1 sports 3 coupe 5 convertible 7.

car\_body\_points(estate, 1).

car\_body\_points(saloon, 2).

car\_body\_points(hatchback, 1).

car\_body\_points(sports, 3).

car\_body\_points(coupe, 5).

car\_body\_points(covertible, 7).

7) Write a rule that uses the facts and the risk\_for\_capacity rule already given to give the total points scored by a car for engine capacity and body type. This will entail 44 Rules adding together the points for the capacity and the points for the body type. The car is to be identified by the manufacturer and model.

total\_points(Man, Mod, Total):-

car(Man, Mod, \_, \_, Cc, Type, \_),

risk\_for\_capacity(Cc, Cpoints),

car\_body\_points(Type, Bpoints),

Total is Cpoints + Bpoints.

Query- total\_points(ford, fiesta, Total).

car(ford,fiesta,popular,uk,950,hatchback,5300).

car(ford,orion,gl,uk,1300,saloon,7800).

car(ford,orion,gl,uk,1600,saloon,8600).

car(ford,orion,ghia,uk,1600,saloon,9500).

car(fiat,uno,55,italy,950,hatchback,5200).

car(fiat,uno,70,italy,1050,hatchback,6500).

car(rover,metro,city,uk,1000,hatchback,4900).

car(rover,metro,mg,uk,1300,hatchback,5700).

car(ford, capri, injection,uk, 2800, sports, 11200).

car(alfa\_romeo, sprint, veloce, italy, 2000, coupe,

12500).

car(volvo, 928, gls, sweden, 1400, hatchback, 6290).

car(mitsubishi, colt, glx, japan, 1800, estate, 7420).

car(mercedes, roadster, 280, germany, 2800, convertible,

18450).

supplier(ford,uk,'21 Tinsgate, Dagenham','081 233 4821').

supplier(rover,uk,'18 Beadle Road, Cowley','0325 24112').

supplier(fiat,italy,'333 Via Alphonse, Turin','0101 888

376 3983').

risk\_for\_capacity(Capacity, 1):-

Capacity < 1000.

risk\_for\_capacity(Capacity, 2):-

Capacity >= 1000,

Capacity < 1300.

risk\_for\_capacity(Capacity, 3):-

Capacity >= 1300,

Capacity < 1500.

risk\_for\_capacity(Capacity, 4):-

Capacity >= 1500,

Capacity < 2000.

risk\_for\_capacity(Capacity, 5):-

Capacity >= 2000,

Capacity < 3500.

risk\_for\_capacity(Capacity, 6):-

Capacity >= 3500.

price\_and\_address(Man, Mod, Trim, Cap, Price, Address):-

car(Man, Mod, Trim, Origin, Cap, \_, Price),

suppliers(Man, Origin, Address, \_).

car\_body\_points(estate, 1).

car\_body\_points(saloon, 2).

car\_body\_points(hatchback, 1).

car\_body\_points(sports, 3).

car\_body\_points(coupe, 5).

car\_body\_points(covertible, 7).

total\_points(Man, Mod, Total):-

car(Man, Mod, \_, \_, Cc, Type, \_),

risk\_for\_capacity(Cc, Cpoints),

car\_body\_points(Type, Bpoints),

Total is Cpoints + Bpoints.