1 Translation of a problem into Prolog

Facts

Blood is red
Summer is hot
Miles is a jazz performer

red(blood).
hot(summer).
jazz_performer(miles).

Properties or relationships by **predicates.**
red, hot and jazz_performer are **predicate names** which denote properties or relations and
blood, summer and miles are **arguments** which denote individuals, or particular occurences.

Note :

Oder of predicates' arguments is arbitrary but MUST be consistent with our interpretation and maintained throughout.

son(john, tom).

son(tom, john).

Important to decide the order of arguments and then strictly adhere to this format.

Lists :

days_of_the_week([mon, tue, wed, thu, fri, sat, sun]).

Order of terms is important.

Multiple Ways of Expression

1. *red(blood).*
2. *colour(blood, red).*
3. *property(blood, colour, red)*

A predicate is determined by its *name* and *arity* :

child(ahmed).
child(ahmed, peter).
child(ahmed, peter, helen).

1. **Ahmed is a child.**
2. **Ahmed is Peter's child.**
3. **Ahmed is a child of Peter and Helen.**

child/2 : an unambiguous specification of the predicate with the name *child* nad with two arguments.

Hence :

child(ahmed, peter).
child([ahmed, peter], helen).
child(helen, 2)

Tom and Ann are pupils. Dr. Richard is their lecturer and Prof. Green is their H-O-D. Relations of this type can be expressed using a tree.

lecturer(dr_richard, tom). lecturer(dr_richard, ann). h_o_d(prof_green, dr_richard).

RULES

A person will buy things if he/she needs them and he/she can pay their price

```
buy( Person, Thing) :-
   needs( Person, Thing),
   price( Thing, Price),
   can_pay( Person, Price).
```

Write a Prolog rule for :

A Day is nice if it is sunny and not windy.

More

X will win a boxing match if X will either knock-out Y or if he/she will get more points than Y

```
win_a_boxing_match( X, Y) :-
  knock_out( X,Y) ;
  (points( X, Px),
   points( Y, Py),
   Px > Py).
```

Write a Prolog rule for :

For all X, Y and Z, X is a grandfather of Y if X is a father of Z, and Z is either a father or a mother of Y.

Lists

Sequences of k $\geq$ 0 terms, enclosed in square brackets and seperated by commas.

e.g. [] (empty list) (k = 0)

2, mary, brother(peter,X)

Q: lst([1,a,X])
A: yes

Q: lst([ ])
A: yes

Q: compound([1,a,X])
A: yes

Q: compound([])
A: no

List is a Prolog structure used to represent an ordered sequence of terms. Also,

([mon,tue,wed,thu,fri,sat,sun])

Elements of a list can be arbitrary Prolog terms

dave(1, date(12, nov 88)),
joey(2, date(15, jul 70)),
tomy(3, date(21, dec 75))

Elements of a list may also be a list

[a,b

, [a], [b], []]