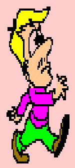
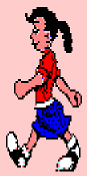
University of Regina **CS 820**

Assignment #2

Total: 100%

Part 1: Knowledge representation in PROLOG (20 %)





We are in a marriage bureau with a file of candidates to the marriage containing the following facts:

• Paul is tall, brown and of age ripe. He likes classical music, the books of adventure and swimming. His future wife must be tall, with red hair and young.

• Kevin is not very tall, fair and young. He likes rock'n'roll, the books of science fiction and tennis. His future wife should be fair, young and not very tall.

• Doug is small, brown and of age ripe. He likes jazz, the detective novels and tennis. He seeks a small, fair woman of average age.

• Alice is not very tall, fair and of average age. She likes any type of music, the books of adventure and swimming. She seeks a tall man, brown and of average age.

• Eva is not very tall, fair and young. She likes rock'n'roll, books of science fiction, and all sports. She seeks a young man, fair and not very tall.

• Lea is small, brown and of age ripe. She likes classical music, the books of adventure and swimming. She seeks a brown man, not very tall and of age ripe.

1. Using PROLOG, express the facts above in the form of predicates.

2. It is considered that two people X and Y, of different sexes, are matched if :

X is appropriate to Y and Y is appropriate to X.

X is appropriate to Y if:

- X is *appropriate physically* to Y (height, color of hair and age of X are those which Y seeks) and,
- X and Y have the same tastes regarding music, literature and sport.

Write a PROLOG program determining the matched couples.

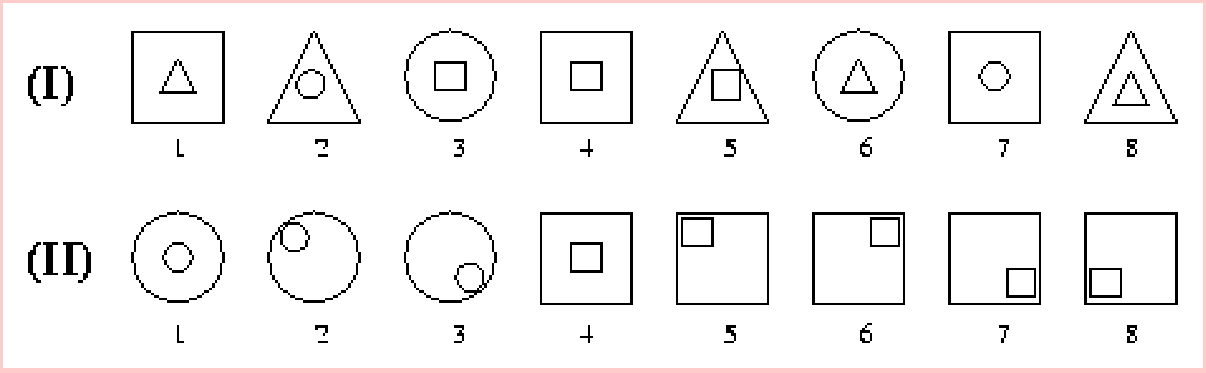
Example 1 :

```
| ?- match(X,Y).

X = kevin
Y = eva ? ;

X = eva
Y = kevin ? ;

no
```

Part 2: The Analogy Problem (20 %)

Consider the task of solving geometric analogy problems, typically used in intelligence tests. A list of figures is given. Three figures A, B, C are singled out and the candidate is asked: " *If figure A is related to figure B, then what figure is C related to ?*".

1. Write a Prolog program to solve this problem.
2. Test the program on the following sets of figures.



- **Example 2 : figures of set (I)**

```
?- analogy1((1,5),(3,X)).
  X = 7
  ?- analogy1((1,5),(2,X)).
  X = 6
  ?- analogy1((1,5),(4,X)).
  X = 4
  ?- analogy1((5,1),(8,X)).
  X = 8
  | ?- analogy1((6,2),(5,X)).
  X = 1 ?
```

- **Example 3 :   figures of set (II)**

```
?- analogy2((1,2),(4,X)).
  X = 5
  ?- analogy2((2,3),(5,X)).
  X = 7
  ?- analogy2((4,7),(1,X)).
  X = 3
  | ?- analogy2((7,5),(3,X)).
  X = 2
  | ?-  analogy2((5,4),(2,X)).
  X = 1
```

Hint:

1. Represent each figure with a fact describing the relationship between its components. For example, figure 1 of set (I) is represented by the fact :

    - **figure(1,middle(triangle,square))**

- Suppose that the question is expressed as: "*If figure number A is related to figure number B, then what figure is figure number C related to ?*". The algorithm to answer this query is as follows:
    - To find the number X of a figure that is related to figure C in the same way as figure B related to figure A do retrieve figures FA, FB, FC numbered A, B, C, then find a rule that relates FA to FB, then apply this rule to FC to obtain a figure FX, and then scan the existing figures to find the number X of FX.

Part 3: Prefixe and postfixe notations (30%)

Using the ADT Stack, write a program in PROLOG to convert an arithmetic expression from postfix(reverse polish notation) to prefix form. The arithmetic expression may contain the 4 arithmetic operators   + , - , / , *   and the unary functions : sin, cos, tan, exp, log and sqrt.

Examples :

```
| ?- post2pre([12,13,+],X).

X = [+,12,13]


| ?- post2pre([12,13,+,4,1,-,/] ,X).

X = [/,+,12,13,-,4,1]


| ?- post2pre([12,13,+,4,1,-,exp,/],X).

X = [/,+,12,13,exp,-,4,1]

| ?- post2pre([30,cos,30,cos,*,30,sin,30,sin,*,+,log,exp,sqrt],Res).

Res = [sqrt,exp,log,+,*,cos,30,cos,30,*,sin,30,sin,30] ?
```

Part 4: Tree Structures + "Evaluation" (30%)

In this assignment, we use a "decision tree" to represent a function on integers: Here a decison tree is a binary tree whose interior nodes are each labeled with a test, such as "x>2", and whose leaf nodes are each associated with a value, such as "9". For now, encode each leaf node as a single number (eg, "9"), and each interior node as a list of three elements [Left, Test,Right], where Left contains the left subtree, Right contains the right subtree, and test is a list [Op, Val], where Op is one of "eq", "lt" or "gt", and Val is a number. (Eg, "[eq,7]" or [gt,2].)

A decision tree represents a function f(x). Given some number x as input, it will compute the value of f(x) as follows: In general, we will first run the test in the root of the tree. If that test succeeds, we take the left branch, and evaluate the tree rooted there, on the given value. If the test fails, we take the right branch. When we eventually reach a leaf, we returns that value.

Hence, to represent the function

```
f(x) =
  if x < -5  then -1
  else
        if x > 8 then +1
        else 0
```

we would use the tree

[-1, [lt, -5], [1, [gt, 8], 0]]

Define the **evalDT(DT, X, V)** predicate that holds if V is the value obtained by evaluating the decision tree DT on the value X. You may assume that the first two arguments are ground, and are well-formed -- eg, the first argument is a legal decision tree and the second is a number.

Example 1:

```
| ?- evalDT([-1,[lt,-5],[1,[gt,8],0]],1,X).

X = 0 ? ;


| ?- evalDT([-1,[lt,-5],[1,[gt,8],0]],-10,X).

X = -1


| ?- evalDT([-1,[lt,-5],[1,[gt,8],0]],12,X).

X = 1 ? ;
```

Example 2:

```
g(x) =
      if x < 0 then
         if x < -10 then -1
         else 0
      else
         if x < 5 then +1
         else +2
```

we would use the tree: **[-1,[lt,-10],0],[lt,0],[1,[lt,5],2]]**

```
| ?- evalDT([[-1,[lt,-10],0],[lt,0],[1,[lt,5],2]],-5,X).

X = 0 ? ;

| ?- evalDT([[-1,[lt,-10],0],[lt,0],[1,[lt,5],2]],3,X).
X = 1


| ?- evalDT([[-1,[lt,-10],0],[lt,0],[1,[lt,5],2]],13,X).

X = 2 ? ;
```

Other examples

evalDT([[-1,[lt,-10],0],[lt,0], [[200, [eq,4],1] , [lt,5],2]] ,4,X).

X = 200 ?

evalDT([-1,[lt,0],[1,[gt,0],0]],40,X).

X = 1 ?

Hand_In :

- Submit via URCOURSES :
    - A prolog file named "**ass2.pl**" and containing the Prolog programs (corresponding to part 1, 2, 3 and 4). As it is mentioned in the above examples, the main predicates corresponding to the programs should be named : **match**, **analogy1** and **analogy2, post2pre and evalDT** respectively.