

# Answers to the mid-term exam on Prolog

Christian Rinderknecht

19 October 2006

## 1 Matching

**Question.** Show the results (Yes/No) and resulting variable bindings for the queries

1.  $?- f(g(X,X), h(Y,Y)) = f(g(Z), Z).$
2.  $?- f(g(X,X), h(Y,Y)) = f(g(h(W,a),Z), Z).$
3.  $?- f(g(X,X), h(_, _)) = f(g(h(W,a),Z), Z).$
4.  $?- f(x(A,B), C) = f(C, x(B,A)).$

**Answers.** When trying to matching two terms, the procedure is:

- if the two terms are constants, they match if they are verbatim the same;
- if one of the terms is a variable, then they match;
- if the two terms are functors, they match if and only if
  - they are the same (name),
  - they have the same number of arguments (subterms),
  - if they have arguments, the  $n$ -th argument of one term matches the  $n$ -th argument of the other.

It is often a good idea to represent the terms to be matched by trees.

1. The figure of the query  $?- f(g(X,X), h(Y,Y)) = f(g(Z), Z).$  is

So the answer is No, because the subterm  $g(X,X)$  does not match  $g(Z)$ : the number of arguments of  $g$  is different;

2. The figure of query  $?- f(g(X,X), h(Y,Y)) = f(g(h(W,a), Z), Z)$  . is

The steps are

$$\begin{array}{llll} X = h(W, a) & X = h(W, a) & h(W, a) = h(Y, Y) & W = Y \\ X = Z & X = h(Y, Y) & X = h(Y, Y) & Y = a \\ Z = h(Y, Y) & Z = X & Z = X & X = h(Y, Y) \\ & & & Z = X \end{array}$$

Finally

$$\begin{array}{l} X = h(a, a) \\ Y = a \\ W = a \\ Z = h(a, a) \\ \text{Yes} \end{array}$$

This answer means that if we take the two trees of the query and replace the variables according to the substitution, we get the exact same tree. Indeed, we get

3. The figure of  $?- f(g(X,X), h(_, _)) = f(g(h(W,a), Z), Z)$  . is



Note that each underscore has been replaced with a unique variable, i.e. different from all the other variables. The steps are now

$$\begin{array}{cccc}
 X = h(W, a) & X = h(W, a) & X = h(W, a) & X = h(\alpha, a) \\
 X = Z & X = Z & Z = X & Z = h(\alpha, a) \\
 h(\alpha, \beta) = Z & h(\alpha, \beta) = h(W, a) & W = \alpha & W = \alpha \\
 & & & \beta = a
 \end{array}$$

Finally

$$X = h(\alpha, a)$$

$$W = \alpha$$

$$Z = h(\alpha, a)$$

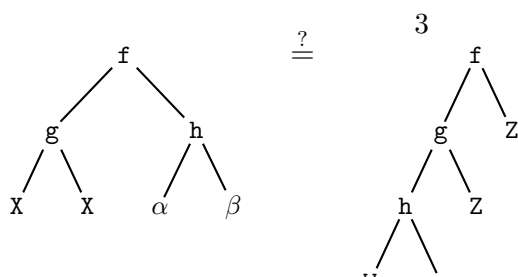
Yes

This means that if we replace the variables in the two initial trees, we get the same. Indeed, we get

4. The figure of  $?- f(x(A,B), C) = f(C, x(B,A)).$  is

The steps are

$$\begin{array}{ccc}
 x(A, B) = C & x(A, B) = x(B, A) & A = B \\
 C = x(B, A) & C = x(B, A) & C = x(B, A)
 \end{array}$$



Finally

This means that if we replace these variables by their value in the original trees, we get the same tree. Indeed, we get

A =  $\alpha$   
B =  $\alpha$   
C =  $x(\alpha, \alpha)$   
Yes

## 2 Mirroring binary trees

A *binary tree* is either a *leaf* or is made of a *root* and two subtrees (left and right). Here roots and leaves are integers. Assume the predicates `leaf(X)`, which denotes a tree made of a leaf X, and `tree(Root, Left, Right)`, which denotes a tree made of a root Root, a left subtree Left and a right subtree Right. Consider `tree(1, tree(5, leaf(3), leaf(4)), leaf(2))` denoting

**Question.** Define `mirror/2` such that `mirror(T1, T2)` means “Binary tree T1 is the mirror of binary tree T2.” As an example, the following trees are mirrors of each other:

```
?- mirror(tree(1, tree(5, leaf(3), leaf(4)), leaf(2)), Mirror).  
Mirror = tree(1, leaf(2), tree(5, leaf(4), leaf(3))) ;  
No
```

**Answer.**

```
mirror(leaf(L), leaf(L)).  
mirror(tree(Root, L, R), tree(Root, MR, ML)) :-  
    mirror(L, ML), mirror(R, MR).
```