

532 Final Exam.

Fall 2009. Time: 2 hours.

1. Assume we give the goal $[[a,b,d],[c],[g],f]=[X|[Y|Z]]$ to a constraint logic programming system. To what are the variables X,Y and Z bound in the result? (10 pts)
2. Consider the labelingsplit predicate defined in page 264 of your textbook. Assume we have a predicate "show(L)" which takes a list of variables, and prints their name, as well as the minimum value that the variable can take, and the maximum value the variable can take. For example, for the goal "[X,Y,Z]::[1..9], X<2, Y>5, show([X,Y,Z])" would print on the screen X 1 1, Y 6 9, Z 1 9, followed by a newline.

Assume we change labelingsplit given in page 264 in the following way:

```
labelingsplit([V|Vs):-  
    show([V|Vs]),  
    mindomain(V,Min),  
    .....
```

What is printed on the screen for the goal "[X,Y]::[6..10], labelingsplit([X,Y])" until the first time the goal succeeds? (10 pts)

3. Consider the definition of "deriv" given on page 300 of your textbook. What is R bound to in the execution of `deriv(mult(plus(3,power(x,2)), mult(4 ,x)), R) ?` (10 pts)

4. Assume the prolog database contains facts like below.

```
parent(ali,veli). /* ali is the parent of veli*/
```

```
parent(fatma, ayse).
```

```
male(ali). /* ali is male */
```

```
female(fatma). /* fatma is female */
```

....

Write predicates for answering queries of the form:

```
grandfather(X,Y) /*X is the grandfather of Y*/
```

```
first_cousin(X,Y) /* X is the first cousin of Y*/
```

```
second_cousin(X,Y) /* X is the second cousin of Y*/
```

```
relative(X,Y) /* X and Y have any kind of blood relationship */ (20 pts)
```

5. Assume the prolog database contains flight information as below.

```
flight(ankara, istanbul, 5600).
```

flight(ercan, izmir, 5600).

6. Write a program defining the predicate `path(S,E,D)` which gives the distance `D`, travelled from start city `S` to end city `E`. (20 pts)

7. Write predicates `put(Q0, I, Q)` and `get(Q0,I,Q)` which implement a queue. The call `put(Q0,I,Q)` should place item `I` at the tail of `Q0` to give `Q`. The call `get(Q0,I,Q)` should return the item `I` at the head of queue `Q0` and the remainder of the queue in `Q`. You should represent a queue simply as a list of elements. (20 pts)

8. Write a predicate for the query `reverse(L,L2)`, which returns the reverse of list `L` in list `L2`. Your predicate should work in $O(n)$ time. (Hint: use an auxiliary predicate, with an accumulator). (10 pts)