CS2LAP: Logic and Prolog 2000/2001	**Handed out:** 23/11/00
Coursework 4	**Due date:** 04/12/00

Important note. Answers to questions 3, 4 and 5 have to be submitted electronically. Put all programs into a single file called `cw4.pl` located in a directory called `cs2lap` under your home directory. The file will be collected automatically at midnight on the due date. If you have any doubts whether you understand and are able to follow these instructions, make sure you ask well *before* the deadline.

Question 1. Consider the following formulas of first-order predicate logic:

$$\begin{array}{ll} (1) & \forall x.\forall y.(A(x,y) \Rightarrow \neg A(x,f(y))) \\ (2) & \forall z.(A(x,y) \Rightarrow B(x,g(x,z)) \vee C(f(z))) \\ (3) & (\exists x.(A(f(x)) \Rightarrow A(y))) \wedge B(x) \end{array}$$

Here x, y, and z are all names of variables (i.e. not constant symbols).

(a) Which variables have a bound occurrence in formula (1)?

(b) Which variables have a free occurrence in formula (3)?

(c) Which terms have a bound occurrence in formula (2)?

(d) Which terms have a free occurrence in formula (2)?

(e) Which (proper) subformulas of formula (2) have a free occurrence?

(f) Which subformulas of formula (3) are in the scope of the existential quantifier?

(g) Is any of the three formulas a sentence? If your answer is 'yes' say which one(s).

Marks: 21 (3 each)

Question 2. Give goal-directed proofs for the following arguments:

(a) $\forall x.(A(x) \Rightarrow B(x)) \vdash (\forall x.A(x)) \Rightarrow (\forall x.B(x))$

(b) $\vdash \exists x.\forall y.(A(x) \Rightarrow A(y))$

Hint: Part (b) is a bit more difficult than it may seem. For a correct solution you'll have to use the restart rule.

Marks: 34 (17 each)

Question 3. **[Electronic submission]** Assume the following operator definitions have been made in Prolog in order to be able to represent formulas of propositional logic.

```
:- op( 100, fy, neg),
   op( 200, yfx, and),
   op( 300, yfx, or),
   op( 400, yfx, implies).
```

Define a predicate `negsf/1` that succeeds iff the argument given represents a logic formula that has at least one negated subformula (including the formula itself). Examples:

```
?- negsf( (p and q) implies (neg (p or q) and neg r)).
 Yes

?- negsf( p and q and (r or s)).
No
```

Marks: 15

Question 4. **[Electronic submission]** Recall that Prolog lists may be interpreted as sets if we ignore ordering and duplicate elements. Without using the built-in predicate `subtract/3`, implement a predicate `diff/3` to compute the difference of two given sets. Example:

```
?- diff( [1, 2, 3, 3, 3, 4, 5, 6, 7, 8, 9, 10], [2, 3, 5, 7], Set).
Set = [1, 4, 6, 8, 9, 10]
Yes
```

Make sure your predicate behaves correctly also when the semicolon key (for alternative solutions to a query) is pressed.

Marks: 15

Question 5. **[Electronic submission]** Write a Prolog predicate `pyramid/1` to print out a pyramid of asterisks of a given height like in the following example.

```
?- pyramid( 7).
      *
     ***
    *****
   *******
  *********
 ***********
*************
Yes
```

Marks: 15