2009 Paper 3 Question 7

Prolog

(*a*) Write a short implementation for each of the following predicates.

(*i*) `unify/2` is true if and only if its two arguments can be unified. [1 mark]

(*ii*) `fail/0` is never true. [1 mark]

(*iii*) `numequal/2` is true if and only if its arguments, interpreted as numerical expressions (assume integer values and no use of division), are numerically equal. For example, `numequal(1+3,2+2)` is true. [1 mark]

(*iv*) `member/2` is true if and only if its first argument is within the list that is its second argument. [1 mark]

(*b*) Given the following code, list all solutions, in order, for the query `c(X,Y,Z)`.

```
a(1).
a(2).
b(a).
c(A,B,C) :- a(A),d(B,C).
c(A,B,C) :- b(A),d(B,C).
d(B,C) :- a(B),!,a(C).
d(B,_) :- b(B).
```

[4 marks]

(*c*) The recursive clause of a bubblesort predicate is reproduced below (assume that `append/3` is defined already). Define *two* different base clauses for this predicate, one of which should use a green cut. When the first argument is unified with a list containing only integers, both of your answers should produce no additional solutions on backtracking. Explain how your complete bubblesort predicate works, including the purpose of the red cut.

```
bubblesort(X,Y) :-
  append(A,[H1,H2|B],X), H1 > H2, !,
  append(A,[H2,H1|B],X1), bubblesort(X1,Y).
```

[6 marks]

(*d*) The power set of a set S is the set of all subsets of S. We will represent sets using lists (ignore list order and assume no duplicates). Write a predicate `ps(+S,-PS)` that unifies `PS` with the power set of `S`, and explain how it works. Include the code for all predicates that you use. [6 marks]