CSCE 580 Spring 2003
SECOND MIDTERM EXAM
Monday 2003-04-14—One Page, Closed Book
35 points, including three extra credit points

1 Prolog—35 points

1. If it exists, give the most general unifier for:
   (a) (2 points) `f(s(s(s(X))))` and `f(s(s(s(s(Y)))))`. **Answer:** {X \ s(Y)}
   (b) (3 points) `knows(john,X)` and `knows(Y,mother(Y))` **Answer:** {Y \ john, X \ mother(john)}
   (c) (3 points) `q(Y,g(a,b)` and `q(g(X,X),Y)` **Answer:** Does not exist
2. (5 points) Write a program for the ancestor relation[1] (called "predecessor" in your textbook.) Assume that the parent relation is already defined. Make sure that the program is both declaratively and procedurally correct. **Answer:** ancestor(X,Y) :- parent(X,Y). ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
3. (5 points) Define the predicate `reverse( List, ReverseList)`, with the obvious meaning. Assume that the predicate `conc(L1,L2,L3)` (list concatenation, also known as `append`) described in class is already defined. **Answer:** rev([],[]). rev([H|T],Z) :- rev(T,T1), append(T1,[H],Z).
4. (5 points, exercise 3.5[B], modified) Define the predicate `palindrome( List)`. A list is a palindrome if it reads the same in the forward and backward direction. For example, `[m,a,d,a,m]`. Assume that `List` contains only constants. **Answer:** palindrome([]). palindrome(L) :- reverse(L, L1), L = L1. Also (Bratko): palindrome([]). palindrome([A]). palindrome(L) :- append([First|Middle], [First], List), palindrome(Middle).
5. (5 points) Define the predicate `delete( Element,List1,List2)` that holds when List2 is List1 with one occurrence of List2 deleted.

   Answer: `del( X,[X|L],L).`
   `del( X,[ Y|Tail],[ Y|Tail1]) :- delete( X,Tail,Tail1).`
6. (5 points) The `append_dl` relation, `append_dl( X-Y,Y-Z,X-Z).`, defines append (viz. concatenate) with difference lists.

[1] For the purpose of this exam, "relation" and "predicate" are used interchangeably.

Explain how it works by showing how terms are matched (viz. unified) for the goal `?-append_dl([a,b|A]-A,[c,d|C]-C,Result).`

Answer: `{X / [a,b|A]}`
`{Y / A}`
`{A / [c,d|C]}`
`{Z / C}`
verb+Result / X-Z+, i.e., `{Result/[a,b|A]-C}`, i.e. `{Result/[a,b,c,d|C]-C}`

7. (5 points, exercise 5.5 [B]) Let S1 and S2 be two lists representing sets. Define a relation `set_difference( X,Y,Z)` that holds if Z is the set difference of X and Y (i.e., Z contains the elements of X that are not in Y). Assume that `member\2` has already been defined. Hint: use cut to distinguish the case in which the first element of the first list is a member of the second list from the case in which the first element of the first list is not an element of the second list.

   Answer: `set_difference( [],_,[]).`

   ```
   set_difference([X|L1],L2,L) :- member(X,L2),!,set_difference(L1,L2,L).

   set_difference([X|L1],L2,[X|L]) :- set_difference(L1,L2,L).
   ```