CSE505 – Fall 2012
Assignment 5: Logic Programming and Prolog
Assigned Weds, Nov 14
Due Weds, Nov 28
Submit online one file called asst5 containing your solution to all three problems.

1 [25%]. Suppose we use Prolog constructors 'lam' and 'app' to represent lambda abstraction and application respectively. For example, we can represent a lambda-term (λf. λx.(f x) (f y)) by the Prolog term app(lam(f, lam(x, app(f, x))), app(f, y)). Write a predicate **occurs_free_in(V, T)** which tests whether a variable V occurs free in a lambda-term T.

2 [40%]. Consider a directed graph represented by the **edge** relation discussed in class. Given as input a start and end node, you should define a **path** predicate that returns in its third argument a path (i.e., a list of nodes) between these two nodes in the directed graph. By repeated backtracking, the path predicate should return all paths.

Test your program using file 'edge' in http://www.cse.buffalo.edu/LRG/CSE505/Prolog/Misc.

As a starting point for your solution, consider the following pair of predicates, path and path2.

```
path(Start, End, Path) :- path2(Start, End, P),   Path  = [Start|P].

path2(X, Y, [Y])    :-  edge(X,Y).
path2(X,Y, [Z|P])  :-  edge(X,Z),  path2(Z,Y,P).
```

The above code works fine for acyclic graphs but it will get caught in a nonterminating loop for cyclic graphs. Modify the code so that this nontermination is avoided and the path predicate works correctly as intended. Hint: Add an extra argument to path2 in which you maintain a list of all nodes that have been visited. Consult this list to avoid traversing cyclic paths.

3 [35%]. Refer to 'xmlbasic' and 'lex' in http://www.cse.buffalo.edu/LRG/CSE505/Prolog/Misc. The test file to be used is 'authors' in the same directory. After loading 'lex'and 'xmlbasic', do:

```
|?- xml2db(authors).
```

The 'author' relation will be asserted. The following question is based upon this relation.

Write a Prolog predicate to determine the oldest author, i.e., the person with the earliest year of birth. You should not assume that the database facts are stored in any particular order. Write two versions of this predicate:

(i) **oldest1**(N), which returns in N the name of the oldest author, using negation-as-failure; and
(ii) **oldest2**(N), which returns in N the name of the oldest author, using setof.

For part (i), you should not use any set collection facilities such as setof, findall, etc.

Suppose the above predicates are saved in a file, called 'queries', test your solution follows:

```
|?- [lex,xmlbasic,queries].
|?- xml2db(authors).
|?- listing.
|?- oldest1(N1), oldest2(N2).
```

On-line Submission: The file 'asst5' to be submitted online should have the definitions for the following predicates: occurs_free_in, path, path2, oldest1, and oldest2.

End of Assignment 5