Programming Project 7

Due date: Tue 5/4 11:00 pm

1 Introduction

In this assignment you will handle simple data and list data with Prolog programs.

2 Paths on Graphs

In this part we work with labeled directed acyclic graphs using the following Prolog representation. The atom \texttt{e}(a,b) indicates that there is an edge from \texttt{a} to \texttt{b} in the graph. Each node in the graph is also given a single color using the appropriate predicates, as in the following example:

\begin{verbatim}
    e(1,2).
    e(2,3).
    e(3,4).
    e(4,5).
    e(1,3).
    e(2,5).
    blue(1).
    green(2).
    black(3).
    red(4).
    black(5).
\end{verbatim}

While we illustrate the requirements for the graph above your program should work for any graph given in this format.

Part I: write a function \texttt{path1}(X,Y) that is true whenever there is a path from \texttt{X} to \texttt{Y} in the graph. For example

\begin{verbatim}
    ?- path1(1,5).
    Yes
    ?- path1(1,3).
    Yes
    ?- path1(3,5).
    Yes
    ?- path1(3,1).
    No
\end{verbatim}

Part II: write a function \texttt{path2}(X,Y) that is true whenever there is a path from \texttt{X} to \texttt{Y} in the graph which \textit{does not} use an edge whose endpoints are colored \texttt{red} \text{-->} \texttt{black}. For example

\begin{verbatim}
    ?- path2(1,5).
    Yes
    ?- path2(1,3).
    Yes
    ?- path2(3,5).
    No
\end{verbatim}
Part III: write a function `path3(X,Y)` that is true whenever there is a path from X to Y in the graph which does use an edge whose endpoints are colored red --> black. For example

?- path3(1,5).
Yes
?- path3(1,3).
No
?- path3(3,5).
Yes

3 Working with Lists

In this part you will work with a database capturing people and the CDs they own. Each CD is given a name and its main artist is listed. The data is given as a list in the following format:

```
[[john,[cd1,clapton],[cd2,moo],[cd3,hip]],
 [janet,[cd3,hip],[cd4,moo],[cd5,u2]],
 [beatrice,[cd6,doors],[cd4,clapton],[cd3,hip]],
 [robert,[cd7,clapton],[cd3,hip],[cd6,doors]]
```

Each element of the main list represents a person. The person’s name is the first element. Following elements are lists of length 2, each giving a CD name and the artist. As in the previous part we illustrate the requirements using this database but your programs should work for any database in this format.

Part I: write a function `similarTaste(Data,P1,P2)` that is true when `Data` is a database as above and `P1`, `P2` are two different people who own at least 2 different CDs in common. Notice that the 2 CDs do not need to appear in the same order in the lists. For the data above this holds for janet and beatrice and for beatrice and robert. You do not need to worry about repeated answers.

You may want to break down the task into simpler tasks. For example you can define a function `ownsCD(Data,Person,CD)` that is true when `Person` owns the `CD`. But this is up to you.

Part II: write a function `sameArtists(Data,P1,P2)` that is true when `Data` is a database as above and `P1`, `P2` are two different people who who have CDs by exactly the same artists. Again the CD lists do not need to be identical - they need to include the same set of elements. For the data above this holds only for beatrice and robert. You do not need to worry about repeated answers.

Here again the structure of the code is up to you. A useful sub-task is a function `personArtists(Data,Person,Artists)` that is true when `Artists` is the list of artists whose CDs are owned by `Person`.

4 Files

For your convenience a seed file for `pp7.pl` including the data given above is available at `/comp/80/files/pp7/`

Submitting your program

Put all the Prolog code in a file `pp7.pl`. Assuming your files are in the current directory on one of our Sun workstations or servers, you should submit by typing

```
provide comp80 pp7 pp7.pl
```