CA208 Logic
Lab 2

Working with structured data and recursion

Aim

The aims of this practical is:

- to become familiar with defining and manipulating structured data in Prolog, and
- to become familiar with recursion.

1. Book Recommender

Use a structured data object to represent a book. The information that describes a book is:

- Title
- Author
- Genre
- Number of pages

The genre should be one of the following {crime, drama, comedy, study, fiction, reference}.

You can represent a library as a list of books.

Write a set of rules for recommending a set of books for:

- Holidays (book should be less than 400 pages and not be a study or reference book).
- Revision (book that is either for study or a reference book with more than 300 pages).
- Literary reading (drama books).
- Leisure (books that either comedy or fiction).

Aside:
Your holiday, revision, literary and leisure predicates should take 2 parameters; a "query" book (with as yet unknown (variable) title and author) and a library (a list of books). It will go through the list and evaluates to true if the book at the head of the list is the appropriate type of book (in which case it will match the title and author of the library book with the "query" book). Typing the list of books in each query will drive you mad! There is a built-in predicate **findall** that has 3 arguements, a generating template, a matching predicate and a list of generate predicates. Whenever

the matching predicate is true it adds a predicate in the list using the generating predicate.

Suppose we define each book as a predicate, e.g.

book(illiad, homer, study, 500).
book(c, Richie, study, 150).
book(nt_bible, sams, reference, 480).
book(monty_python, cleese, comedy, 300).

Then we can build the library using

buildLibrary(Lib) :- findall(book(Title, Author, Genre, Size), book(Title, Author, Genre, Size), Lib).

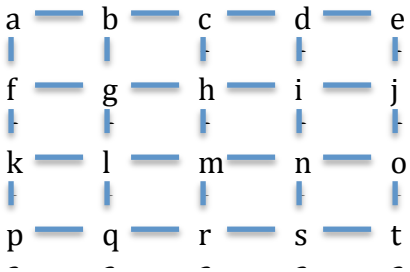
And query the library using

buildLibrary(L), leisure (B,L).

If there are more the one "leisure" book in the library, make sure your leisure predicate can find them when you use the semi-colon to find the next possible answer!

2. North-South-East-West

Consider the following grid of points.



Write a set of facts that represents this grid.

Write a set of rules that tests if a point on the grid is due north, north-west, west, south-west, south, south-east, east or north-east of another point.