EECS 3401 3.0 Intro to AI and LP
Fall 2017

Dept. of Electrical Eng. & Computer Sci.
York University

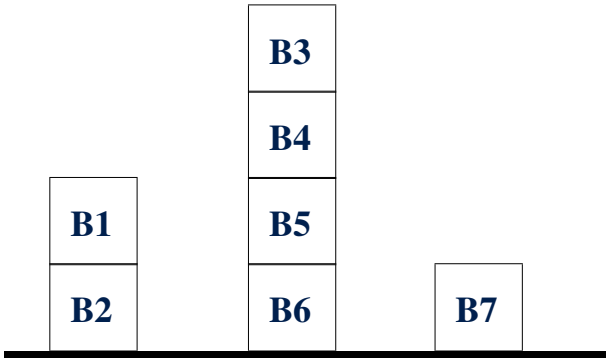
Assignment 1

Total marks: 80.

Out: October 4
Due: October 18 at 14:30

Note: Your report for this assignment should be the result of your own individual work. Take care to avoid plagiarism ("copying"). You may discuss the problems with other students, but do not take written notes during these discussions, and do not share your written solutions.

1. [20 points] Suppose that we represent a 2D scene involving stacked blocks on a table using the two Prolog predicates `on(X,Y)`, meaning that block `X` is directly on block `Y`, and `just_left(X,Y)`, meaning that block `X` and block `Y` are on the table and that `X` is immediately to the left of `Y` . For example, suppose that we have the following scene:



This scene would be represented by the following facts

```
on(b1,b2).   on(b3,b4).   on(b4,b5).   on(b5,b6).
just_left(b2,b6). just_left(b6,b7).
```

Write and test a Prolog program `q1.pl` that implements the following predicates for querying such a representation of scenes:

- `above(X,Y)`, which holds if block `X` is somewhere above block `Y` in the pile where Y occurs. For example, `above(b4,b6)` and `above(b5,b6)` hold, while `above(b3,b1)` does not hold.

- `left(X,Y)`, which holds if block `X` is somewhere to the left of block `Y`, but perhaps higher or lower than `Y`. For example, `left(b2,b6)`, `left(b2,b7)`, `left(b1,b7)`, `left(b2,b3)` hold, while `left(b3,b4)` does not hold.
- `right(X,Y)`, which should hold whenever `left(Y,X)` holds.

Test your program. Submit these test results in the file `q1tests.txt`, together with the program file. Document your code appropriately.

2. [20 points] Write and test a Prolog program `q2.pl` that implements the following list manipulation predicates:

   - `intersect(L1,L2)`, which holds if `L1` and `L2` are lists with an element in common. For example `intersect([1,2,3,4],[5,4,1,6])` holds, but `intersect([1,2,3,4],[5,6])` does not hold. Do not use any built-in predicates other that `member` for this.
   - `all_intersect(L1, L)`, which holds if every element of list `L1` is a list `L3` such that `intersect(L3, L2)` holds. For example, `all_intersect([[1,2,3],[5,4,6]],[3,4])` holds, and `all_intersect([],[3,4])` holds, but `all_intersect([[1,2,3],[1,2,5],[5,4,6]],[3,4])` does not. Use `intersect` and recursion to define `all_intersect`.
   - `member_nl(A,L)`, which hold if `A` is an atom that appears somewhere in the possibly nested list `L`. For example, `member_nl(c,[a, [b, e, f], [e, [g , c, d], b]])` holds but `member_nl(h,[a, [b, e, f], [e, [g , c, d], b]])` does not. `member_nl(X,[a, [b, e, f], [e, [g , c, d], b]])` should repeatedly succeed with `X` bound to the different elements occurring in the nested list, `a`, `b`, `e`, ...

   Test your program. Submit these test results in the file `q2tests.txt`, together with the program file. Document your code appropriately.

3. [20 points] Consider the following puzzle:

   Donna, Danny, David, and Doreen were seated at a tablle in a restaurant. The men sat across facross from each other, as did the women. They each ordered a different main course with a different beverage. In addition,

   (a) Doreen sat beside the person who ordered steak.

   (b) The chicken came with a coke.

   (c) The person with the lasagna sat across from the person with milk.

(d) David never drinks coffee.

(e) Donna only drinks water.

(f) Danny could not afford to order steak.

Who ordered the pizza?

Write and test a Prolog program `q3.pl` that solves the puzzle and displays who ordered each of the main courses and each of the beverages. Use the technique shown in the zebra example discussed in class (the code is available on the course web page) to find the missing information. Begin by writing clauses defining predicates `beside(X,Y)` which holds if person `X` is sitting beside person `Y`, and `across(X, Y)` which holds if person `X` is sitting across from person `Y`. (Encode the above constraints as given and do not add additional facts.)

Test your program. Submit these test results in the file `q3tests.txt`, together with the program file. Document your code appropriately.

4. [20 points] This non-programming question concerns the following situation:

> Joe, Sally, Bill, and Ellen are the only members of the Elm Street Bridge Club. Joe is married to Sally. Bill is Ellen's brother. The spouse of every married person in the club is also in the club.

From these facts, most people would be able to determine that Ellen is not married.

a) Write some sentences in first-order logic that represent the facts given above. Also provide a glossary where you indicate the intended meaning of your predicate, function, and constant symbols in English.

b) Show that the given facts do *not* entail that Ellen is not married, i.e., give an interpretation that satisfies the facts but falsifies the conclusion.

c) Write some sentences in first-order logic that represent the additional knowledge that most people would have and prove that the augmented set of sentences now entails that Ellen is not married. Your proof should use the definition of entailment and refer to interpretations; do not use resolution.

Submit your answer to this question as PDF file `q4.pdf`.

To hand in your report for this assignment, put all the required files in a directory `a1answers` *and submit it electronically by the deadline. To submit electronically, use the following Prism lab command:*

```
submit 3401 a1 a1answers
```

Your Prolog code should work correctly on Prism.