# Midterm 330-001 Fall 2019

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(on back as well)

1. For the following KB, what are the queries for a,b,c, and d. It is fine if your query “returns” (sets) variables; that does not change the success or failure of a query if that is all that is requested.

**%book( stock# , title, edition, year). – stock # is a unique identfier**

**book( id1942, 'The C Programming Language', 1, 1978 ).**

**book( id2491, 'The C Programming Language', 2, 1988 ).**

**book( id9791, 'Pride and Prejudice' ,1 , 1813 ).**

**%... [remember, “…” means “many more like this”]**

**%author( stock#, fname, lname).**

**author( id1942, brian , kernighan).**

**author( id1942, dennis, ritchie).**

**author( id9791, jane, austen).**

**%...**

**%in\_stock(stock#, yes/no)**

**in\_stock( id1942 , no ).**

**in\_stock( id2491, yes).**

**In\_stock( id9791, no).**

**%...**

* 1. (5) **Query** for “is there a book titled “Learn You a Haskell for Great Good”?”

**?-**

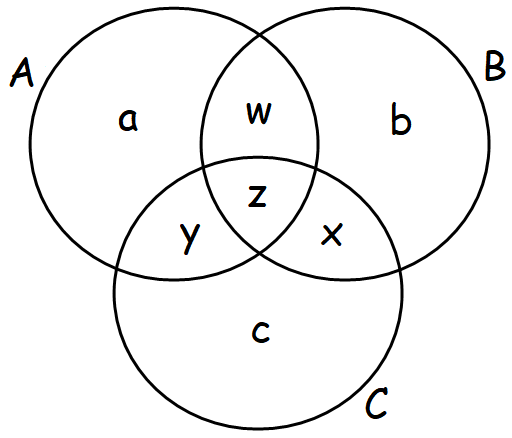
* 1. (5) **Query** for “Are there any books without authors?”

**?-**

* 1. (5) **Query** for “are there any books from this year (2019) not in stock?”

**?-**

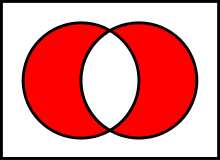
1. Graphs can be represented multiple ways in Prolog.
   1. (5) Give a reasonable way of representing the following Venn Diagram in Prolog. Please consider parts b and c when deciding how to represent. Assume there are many more elements of Sets A,B, and C, so show just a,b,c,w,x,y,z in your representation.



* 1. (5) Define an **intersection** rule, that holds is M is a member of the intersection of two sets, meaning it is a member of each set.

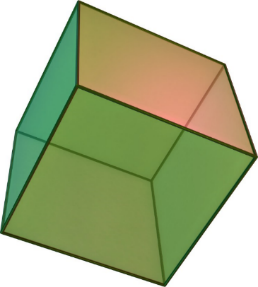
**intersection(A,B,M):-**

* 1. (5) Define a disjunctive union rule that holds if M is a member of A or B but not both. Pictorially, red/gray area, on the right:



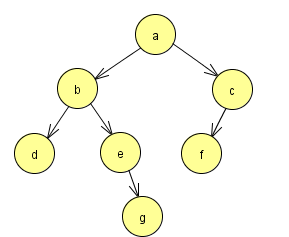
**disj\_union(A,B,M) :-**

1. (15) Map Coloring: Solve the map coloring problem for a Cube. You are to color the faces with at most 3 colors. Opposite faces must have the same color.



**color(red). color(blue). color(green). color(yellow).**

1. Trees, in this case binary trees, are recursive structures, so naturally it’s easy to work with them in Prolog. (these are not binary *search* trees).

****

**%EXAMPLE tree**

**node(a,b,c).**

**node(b,d,e).**

**node(c,f,nil).**

**node(d,nil,nil).**

**node(e,nil,g).**

**node(f,nil,nil).**

**node(g,nil,nil).**

* 1. (5) Define a **valid\_binary\_tree** rule. You may assume every node is connected to the tree (remember that every node in a binary tree can have at most two children and at most one parent). Helper rules are fine, as always.

**valid\_binary\_tree :-**

* 1. (10) Define a **right\_of(Right,Node)** that returns true if **right** is a node in one of the **Node**’s right subtrees. Hint: Compare top-down to bottom-up approaches.

1. (10) [Extra Credit] Write a **between(List,B,X,Y)** rule that holds if a list List contains **B,X**, and **Y** and **B** lies between **X** and **Y**.
2. (4) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ are general features of successful languages
3. (2) Java uses \_\_\_\_\_\_\_\_\_\_\_\_ binding to associate a variable with its type.
4. (3) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is an example of a declarative language.
5. (3) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is an example software development lifecycle model that is used to carry a program from initial requirements all the way through development and testing and eventually release, possibly followed by further updates.
6. (3 ) A language having many features, such as loop statements and ways of performing addition (for a low-level machine language in particular), that are very similar can be said to lack \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
7. (3) The lexemes/tokens of a grammar are generally specified with \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ language(s), the lowest level of the Chomsky hierarchy.
8. (3) Order of operation rules, really precedence and associativity for programming languages, are handled by the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_analysis stage of compilation.
9. (3) The input to the lexer, the program that perfoms lexical analysis, is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
10. (6) Give the parse tree for **bdgxggxg** from **A**,as the start symbol. The alphabet is .

**A->BG**

**B->bB | C**

**C->cC | D**

**D-> dD | ε // ε is “nothing”, the empty string**

**G->xGx | gGg | // ε is “nothing”, the empty string**

1. (5) Give EBNF for a string having any number of **x**s, **y**s, and **z**s but having **xy** and/or **yz**  as a substring.
   1. Positive Examples: **xxyyz, zyxy, xyzzyxzy, yzyyxx**
   2. Negative Examples: **xzyx**, **xxxxx**, **zyxxzx**, **ε**
2. (5) Match the EBNF on the left to the strings they match on the right.

**xyzzzyx**

**(x|y){y|z}x**

**zyxxxyz**

**{x|y|z}x(x|z)**

**yxzyxzyx**

**{x|y|z}z{x|y}**

**xxxx**