

# COMP 3007

•

## Prolog!

**Due: Monday April 4th @ 11:55pm (No Lates!)**

### Question 1

[15 marks] Establish a database of facts for the following 6 predicates:

- `male(X)` % X is male
- `female(X)` % X is female
- `parent(X,Y)` %X is the parent of Y
- `father(X,Y)` %X is the father of Y
- `mother(X,Y)` %X is the mother of Y
- `married(X,Y)` %X is married to Y

Write prolog rules to define the following 10 relationships:

- `different(X,Y)` %X and Y are different
- `is_mother(X)` % X is a mother
- `is_father(X)` % X is a father
- `aunt(X,Y)` % X is an aunt of Y
- `uncle(X,Y)` % X is an uncle of Y
- `sister(X,Y)` % X is a sister of Y
- `brother(X,Y)` % X is a brother of Y
- `grandfather(X,Y)` % X is a grandfather of Y
- `grandmother(X,Y)` % X is a grandmother of Y
- `ancestor(X,Y)` % X is an ancestor of Y

Please assume "traditional" definitions for the above relationships.

If you are in any doubt, include your assumptions in your assignment documentation.

### Question 2

[15 marks total] Given the following database of facts:

```
actor(jonny, depp, gender(male)).
actor(bruce, willis, gender(male)).
actor(glenn, close, gender(female)).
actor(orlando, bloom, gender(male)).
```

```

actor(jennifer, lawrence, gender(female)).
actor(sean, bean, gender(male)).
actor(angelina, jolie, gender(female)).
actor(keira, knightley, gender(female)).
actor(benedict, cumberbatch, gender(male)).

movie(year(2003), title([pirates,of,the,carribean]), cast([actor(jonny,
depp), actor(keira, knightley), actor(orlando, bloom)]))).
movie(year(2001), title([lord,of,the,rings]), cast([actor(orlando, bloom),
actor(sean, bean)]))).
movie(year(1988), title([die,hard]), cast([actor(bruce, willis)]))).
movie(year(2014), title([the,imitation,game]), cast([actor(benedict,
cumberbatch), actor(keira, knightley)]))).
movie(year(2012), title([the,hunger,games]),
cast([actor(jennifer,lawrence)]))).

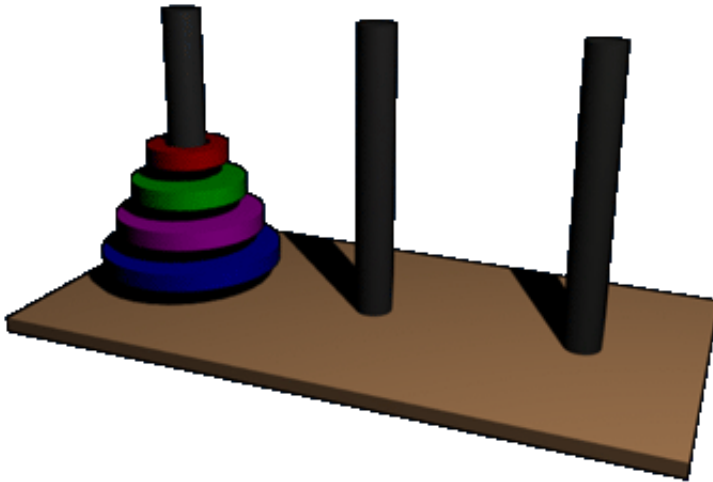
```

Write prolog queries that answer the following questions:

1. [1 mark] What movies were released in or after 2003?
2. [1 mark] What movie(s) contain the word "of" in the title?
3. [2 marks] What movies share one or more common words in their titles?
4. [1 mark] What are the names of the female actors?
5. [2 marks] In what movies were female actors members of the cast?
6. [2 marks] In what movies is Orlando Bloom a member of the cast?
7. [3 marks] What actor(s) are in the cast of more than 1 movie?
8. [3 marks] What actor(s) are not in the cast of any movie?

Hint: You may want to use the member/2 predicate in some of your answers. **For submission, include your queries as comments in the database, together with your testing output.**

### Question 3



[20 marks] Write and test a program to solve the Towers of Hanoi problem. The problem is to move  $N$  disks from the left peg to the right peg using the center peg as an auxiliary holding peg. At no time can a larger disk be placed upon a smaller disk. The following diagram depicts the setup for  $N=3$  disks. The tower problem can be solved recursively using the following strategy. To move  $n$  disks from one peg to another, using a 3rd peg as a holding peg, move  $n-1$  disks to the holding peg, then move the last disk to the destination peg, then move all  $n-1$  disks from the holding peg to the destination peg. Your solution should print out each movement and have the form:

`hanoi(N,Left,Right,Center) /*move N disks from the right peg to the left peg using the center as a holding peg.*/`

#### **Question 4**

[15 marks total]

1. [3 marks] Write a predicate to find the last element of a list. You may not use the built in last predicate in your answer. E.g.,

```
?- mylast(X,[how,are,you,today]).
X=today.
```

2. [3 marks] Write a predicate `after(X,List,Result)` that returns everything in a list after any occurrence of the given element  $X$ . E.g.,

```
?- after(a, [b,a,x,d,a,f,g], R).
R = [x,d,a,f,g].
R = [f,g].
```

3. [4 marks] Write a predicate `nextto(X, Y, L)`, that succeeds when elements  $X$  and  $Y$  are immediately consecutive elements of a list  $L$ . E.g.,

```
?- nexttto(a,b, [c,a,b,d]).
True.
?- nexttto(a,b, [c,a,d,b]).
False
```

4. [5 marks] Write the predicate `occurs_at_position(Element, List, Position)`, that allows access to the *n*th element in a list. E.g.,

```
?- occurs_at_position(x, [a,b,c,x,x,d,e], 4).
True
?- occurs_at_position(x, [a,b,c,x,x,d,e], Pos).
Pos = 3;
Pos = 4;
?- occurs_at_position(X, [a,b,c,x,x,d,e], 2).
X = c
```

### **Question 5**

[10 marks total]

1. [2 marks] Write a predicate `myAppend(L1,L2,L3)` to append 2 lists (order is important, you may not use the built in `append`). E.g.,

```
?- myAppend([a,b],[c,d],L).
L = [a,b,c,d].
```

2. [2 marks] Rewrite the predicate `myLast(X,L)` from the previous question using `append`.  
3. [3 marks] Rewrite the predicate `nexttto(X,Y,L)` from the previous question using `append`.  
4. [3 marks] Write a predicate `myReverse(L1,L2)` that succeeds when the result of reversing the elements of list *L1* is the list *L2*. Your answer should use `append`, and may not make use of the built in `reverse` predicate.

### **Question 6**

[15 marks total] Suppose we represent a directed graph by an edge relation:

```
edge(X,Y,L) %the edge from X to Y is of length L.
```

**Therefore, a graph that looks like a square of size 2 can be defined as follows:**

```
edge(a,b,2).
edge(b,c,2).
edge(c,d,2).
edge(a,d,2).
```

1. [3 marks] Write and test the clauses for the relation:

`connected(X,Y)` %there is a path from X to Y in the graph

E.g.,

```
?- connected(a,c).  
True
```

2. [4 marks] Write and test the clauses for the relation:

`pathLength(X,Y,R)` %the length of the path from X to Y is R

E.g.,

```
?- pathLength(a,d,R).  
R = 6;  
R = 2;
```

3. [4 marks] Write and test the clauses for the relation:

`listForPath(X,Y,L)` %the path between X and Y is the list L.

E.g.,

```
?- listForPath(a,d,L).  
L = [edge(a,d,2)];  
L = [edge(a,b,2),edge(b,c,2),edge(c,d,2)];
```

4. [4 marks] Write and test the clauses for the relation:

`onPath(X,Y,E)` % the edge E is on the path between X and Y

E.g.,

```
?- onPath(a,c,edge(b,c,_)).  
True  
?- onPath(a,c,X).  
X = edge(a,b,2);  
X = edge(b,c,2);
```

**You should define other graphs for testing, using different combinations of edges. Include the definition of the graphs that you tested in your .pl files. Documentation & Testing [10 marks]**

Ensure that your name and student number are in comments at the top of all files. Document the purpose of each method including its expected inputs (parameters) and output (return).

It is up to you to make a convincing, documented test plan. There are standard tests that must be done (base case, testing of some upper or lower bounds of the input or results, etc.) but it is up to you to come up with a sufficient testing scheme.

Include evidence of testing your code either in comments at the bottom of your file or in a separate

testing.txt file. Comment your testing as to what you are testing and why, giving expected output as well as observed output and explanations for any differences.

Fabricated test output will result in 0 marks for a question.

Include a readme file with any instructions for your code, including any major assumptions you have made.

Any files that are not loadable will result in a mark of 0 for that question. Ensure that your code is well-formatted and easily readable; a happy TA is a generous TA.

## Submission

- Submit your assignment using [cuLearn](#).
- If you used multiple files, combine them into a single zip file for the whole assignment.
- Marks will be deducted for late submissions.
- If your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late, so do not wait until the last minute.
- You are responsible for submitting all files and ENSURING that the submission is completed successfully. If you are having issues with submission, contact me **before** the due date.