MINISTRY OF HIGH EDUCATION

Obour High Institute

For Management & Informatics

معاهد العبور
Obour Institutes

FINAL EXAM

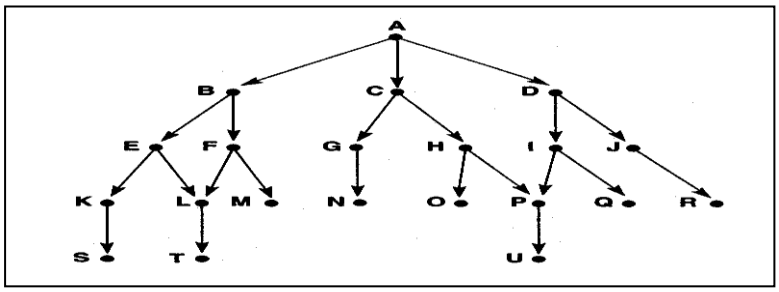
Course Name: Artificial Intelligence	Term: First Academic Year: 2018-2019
Course Code: CS6411	Date: 6/1/2019
Intake & Major: Computer Science	Time: 2 Hours

Answer the following questions

Question One:

Write and trace the algorithms of Backtrack, Depth-first and Breadth-First search by using the following graph. Begin from state A to goal R.

Keep track of the successive value of NSL, SL, CS and DE.



Where SL for state list, NSL for new state list, CS for current state and DE for dead ends.

```
function backtrack;

begin
  SL := [Start]; NSL := [Start]; DE := [ ]; CS := Start;          % initialize:
  while NSL ≠ [ ] do                          % while there are states to be tried
    begin
      if CS = goal (or meets goal description)
        then return SL;              % on success, return list of states in path.
      if CS has no children (excluding nodes already on DE, SL, and NSL)
        then begin
          while SL is not empty and CS = the first element of SL do
            begin
              add CS to DE;                        % record state as dead end
              remove first element from SL;                          %backtrack
              remove first element from NSL;
              CS := first element of NSL;
            end
          add CS to SL;
        end
        else begin
          place children of CS (except nodes already on DE, SL, or NSL) on NSL;
          CS := first element of NSL;
          add CS to SL
        end
    end;
    return FAIL;
end.
```

```
function breadth_first_search;

begin
   open := [Start];                                              % initialize
   closed := [ ];
   while open ≠ [ ] do                                           % states remain
      begin
         remove leftmost state from open, call it X;
            if X is a goal then return SUCCESS                  % goal found
               else begin
                  generate children of X;
                  put X on closed;
                  discard children of X if already on open or closed;   % loop check
                  put remaining children on right end of open           % queue
               end
      end
   return FAIL                                                   % no states left
end.
```

1. **open = [A]; closed = [ ]**
2. **open = [B,C,D]; closed = [A]**
3. **open = [C,D,E,F]; closed = [B,A]**
4. **open = [D,E,F,G,H]; closed = [C,B,A]**
5. **open = [E,F,G,H,I,J]; closed = [D,C,B,A]**
6. **open = [F,G,H,I,J,K,L]; closed = [E,D,C,B,A]**
7. **open = [G,H,I,J,K,L,M]** (as L is already on open); **closed = [F,E,D,C,B,A]**
8. **open = [H,I,J,K,L,M,N]; closed = [G,F,E,D,C,B,A]**
9. and so on until either U is found or **open** = [ ]

```
begin
   open := [Start];                                              % initialize
   closed := [ ];
   while open ≠ [ ] do                                           % states remain
      begin
         remove leftmost state from open, call it X;
         if X is a goal then return SUCCESS                      % goal found
            else begin
               generate children of X;
               put X on closed;
               discard children of X if already on open or closed;   % loop check
               put remaining children on left end of open              % stack
            end
      end;
   return FAIL                                                   % no states left
end.
```

1. **open = [A]; closed = [ ]**
2. **open = [B,C,D]; closed = [A]**
3. **open = [C,D,E,F]; closed = [B,A]**
4. **open = [D,E,F,G,H]; closed = [C,B,A]**
5. **open = [E,F,G,H,I,J]; closed = [D,C,B,A]**
6. **open = [F,G,H,I,J,K,L]; closed = [E,D,C,B,A]**
7. **open = [G,H,I,J,K,L,M]** (as L is already on open); **closed = [F,E,D,C,B,A]**
8. **open = [H,I,J,K,L,M,N]; closed = [G,F,E,D,C,B,A]**
9. and so on until either U is found or **open** = [ ]

Question Two:

Write and trace of the execution of best_first_search by using evaluation function f(n) for following state space generated in heuristic search of the 8-puzzle graph.

Consider the evaluation function f(n)=g(n)+h(n),

where n is any state encountered in the search.

g(n) is the cost of n from the start state.

h(n) is the heuristic estimate of the cost of going from n to the goal.

What is the role of the g(n), answer by another trace with f(n)=h(n).

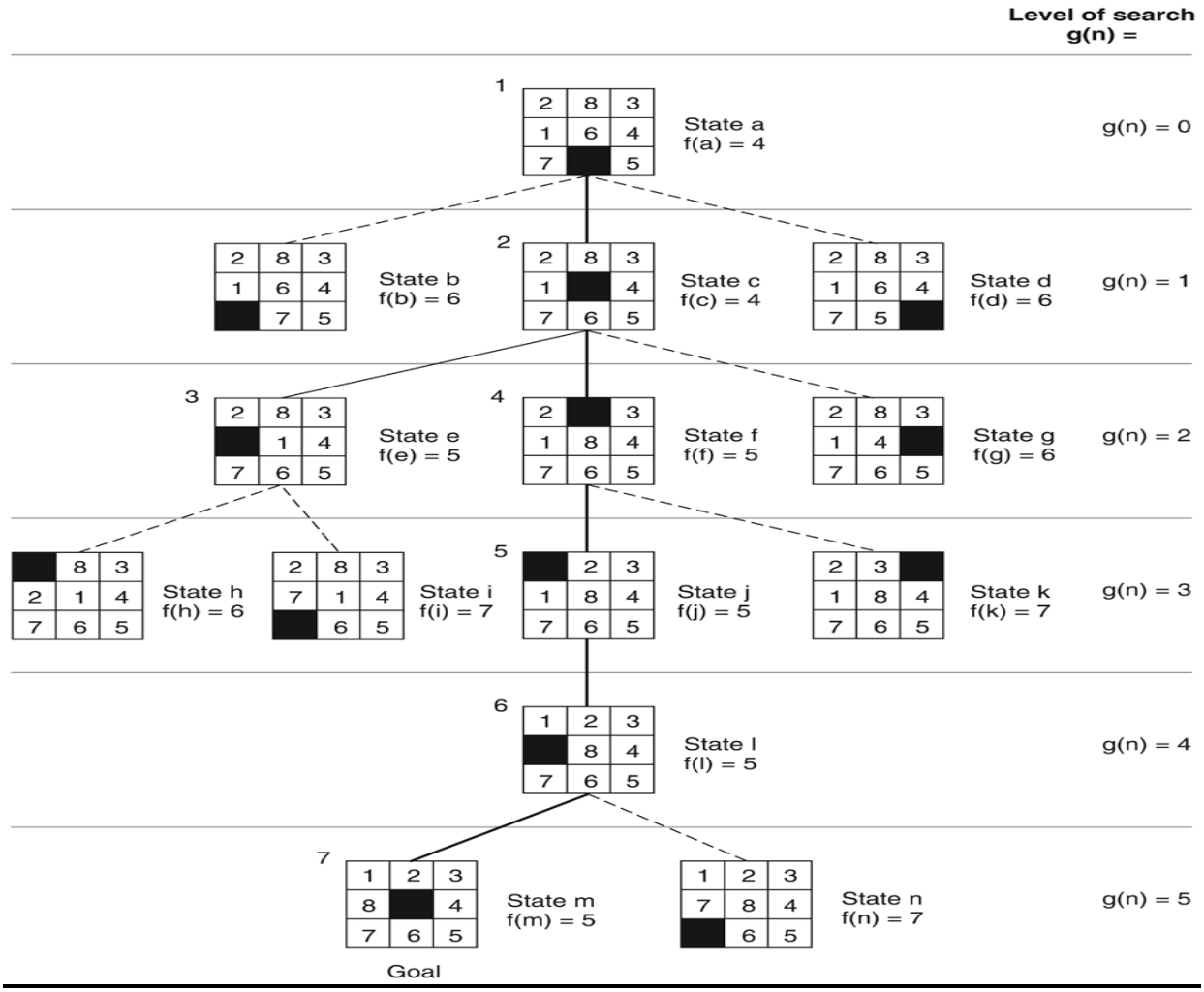
Write your comments.

2	8	3
1	6	4
7		5

Start State

1	2	3
8		4
7	6	5

Goal State



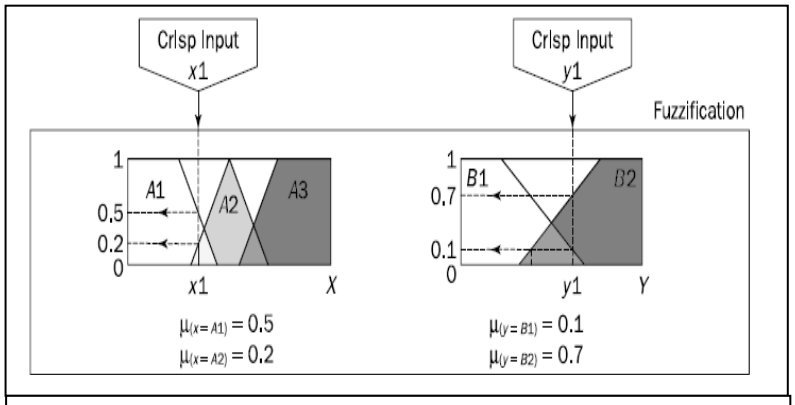
Question Three:

Draw the basic structure (Mamdani-style) that simulate the Fuzzy inference(Fuzzification, Rule evaluation, Aggregation of rule consequents, Defuzzification) for the following rules

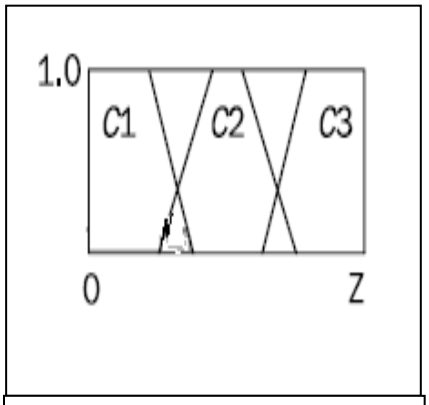
1. *IF project_funding is adequate*
   OR project_staffing is small
   THEN risk is low
2. *IF project_funding is marginal*
   AND project_staffing is large
   THEN risk is normal
3. *IF project_funding is inadequate*
   THEN risk is high

Suppose the ranges of project funding and project staffing between 1 to 100 per cent.
And the crisp input x1=0.35 and y1=0.6

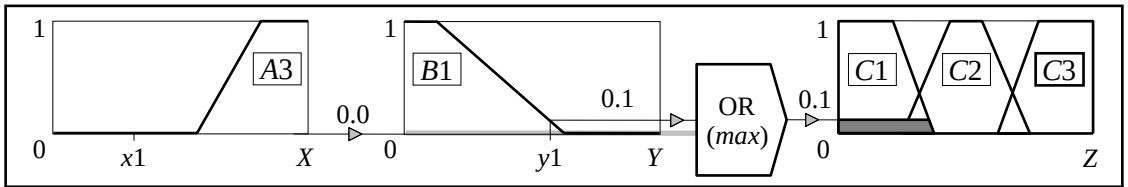
The membership degree for project funding and project staffing and risk as follow:



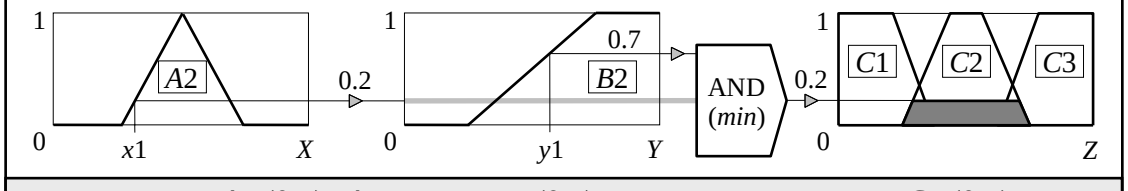
The membership degree for project funding and project staffing



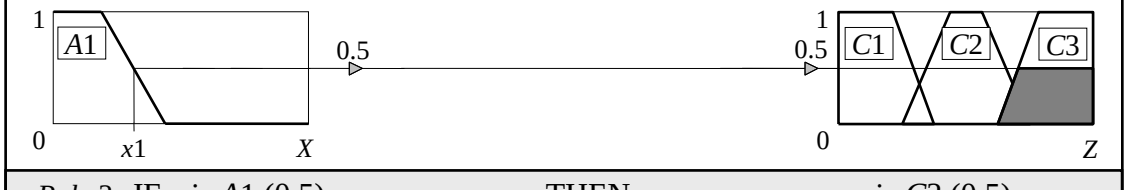
The membership degree for risk



Rule 1: IF *x* is *A*3 (0.0) OR *y* is *B*1 (0.1) THEN *z* is *C*1 (0.1)

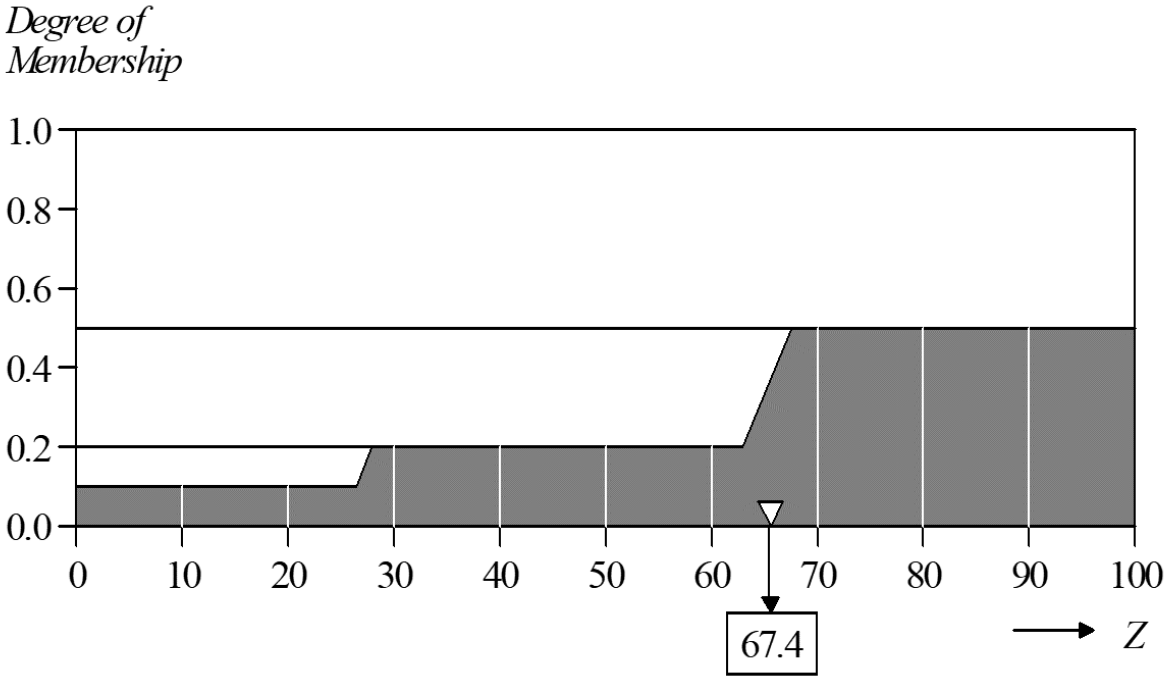


Rule 2: IF *x* is *A*2 (0.2) AND *y* is *B*2 (0.7) THEN *z* is *C*2 (0.2)



Rule 3: IF *x* is *A*1 (0.5) THEN *z* is *C*3 (0.5)

$$COG = \frac{(0+10+20)\times 0.1+(30+40+50+60)\times 0.2+(70+80+90+100)\times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$



Question Four:

a. Define the ontology.
b. How to develop an ontology.
c. Consider the following tabular data (bibliography) and answer the following questions.

Books					
Id	Title	Author	Publisher	Category	ISBN
1	Introduction to computers	Jim Hendler	springer	Semantic Web	978-0-12-385965-5
2	Essential bioinformatics	JIN XIONG	springer	Bioinformatics	978-0-470-02001-2
3	Pattern discovery in bioinformatics	David L. Olson	springer	Data Mining	978-3-540-76916-3
4	Advanced databases	Borko Furht	springer	Cloud Computing	978-1-4419-6523-3
5	Algorithms of bioinformatics	Frédéric Dardel	springer	Bioinformatics	978-0-470-12321-2

1. Convert the tabular data into XML formats where the "id", "title", "author", "publisher", "category" and "ISBN" are attributes for the element book and the root element is library.
2. Convert the tabular data into the RDF representation using the following:
   i. The global URI for the rdf namespace is http://www.w3.org/1999/02/22-rdf-syntax-ns#.
   ii. "bib" stands for http://www.amazon.com/books-used-books-textbooks.
   (Convert 2 records only).

a. Define the ontology.

An **ontology** is a formal explicit description of concepts in a domain of discourse (**classes** (sometimes called **concepts**)), properties of each concept describing various features and attributes of the concept (**slots** (sometimes called **roles** or **properties**)), and restrictions on slots (**facets** (sometimes called **role restrictions**)).

An ontology together with a set of individual **instances** of classes constitutes a **knowledge base**. In reality, there is a fine line where the ontology ends and the knowledge base begins.

b. How to develop an ontology

defining classes in the ontology
arranging the classes in a taxonomic (subclass–superclass) hierarchy,
defining slots and describing allowed values for these slots,
filling in the values for slots for instances.

c. 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
<book>
<ID> 1 </ID>
<title> Introduction to computers</title>
<author>Jim Hendler</author>
<publisher>springer</publisher>
<category>Semantic Web</category>
<ISBN>978-0-12-385965-5</ISBN>
</book>
<book>
<ID> 2 </ID>
<title>Essential Bioinformatics</title>
<author>JIN XIONG</author>
<publisher>springer</publisher>
<category> Bioinformatics</category>
<ISBN>978-0-470-02001-2</ISBN>
</book>
</library>
```

```
2.<rdf:RDF
xmlns:bib="http://www.amazon.com/textbooks#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntaxns#"
<bib:Book rdf:about="http://www.amazon.com/textbooks#Book1">
<bib:id>1</bib:id>
<bib:title> Introduction to computers </bib:title>
 <bib:author> Jim Hendler</bib:author>
<bib:publisher> springer</bib:publisher>
<bib:category> Semantic Web</bib:category>
<bib:ISBN>978-0-12-385965-5</bib:ISBN>
</bib:Book>
<bib:Book rdf:about="http://www.amazon.com/textbooks#Book2">
<bib:id>2</bib:id>
<bib:title>Essential Bioinformatics</bib:title >
<bib:author> JIN XIONG</bib:author>
<bib:publisher> springer</bib:publisher>
<bib:category> Bioinformatics </bib:category>
<bib:ISBN>978-0-470-02001-2</bib:ISBN>
</bib:Book>
 </rdf:RDF>
```

Question Five:

By using the tabular data in the question 3 answer the following:

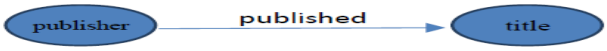
a. By using the graph representation, represent the relation between
- i. The publisher and the title of the book (published)
- ii. The title and the author of the book (wroteby)
- iii. Merge the two graphs in one graph.

b. Write SPARQL query to determine the following
- i. The books which published by the springer.
- ii. The publisher of the book "Pattern Recognition"
- iii. Write the answer of the following query:

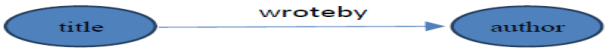
```
SELECT ?who
WHERE {:springer :published ?what .
       ?what :wroteby ?who .}
```

a. By using the graph representation, represent the relation between

A. The publisher and the title of the book (published)



B. The title and the author of the book (wroteby)



C. Merge the two graphs in one graph.

b.

i. The books which published by the springer.

Representation:

SELECT?what
WHERE {:Springer :Published **?what.**}

ii. The publisher of the book "Pattern Recognition"

SELECT?who
WHERE{: Pattern Recognition:PublishedBy **?who.**}

iii. Write the answer of the following query:

SELECT?who
WHERE {:springer :published**?what**
?what:WroteBy**?who.**}
RESULT

Author	Title
Jim Hendler	Bioinformatics Genomics& post genomics
JIN XIONG	Introduction of bioinformatics
David L. Olson	Introduction to Semantic Web
Borko Furht	Advanced Data Mining Techniques
Frédéric Dardel	Handbook of Cloud Computing