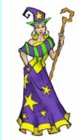
CSE4403 3.0 & CSE6002E - Soft Computing
Winter Semester, 2011



Final Exam

Date: 14 April 2011 (in 1016 Vari Hall, 9:00-10:30)

1. Logic (15 points)
   (a) 3
   (b) 3
   (c) 6
   (d) 3
2. Neural Networks (15 points)
   (a) 5
   (b) 8
   (c) 2
3. Fuzzy Logic (15 points)
   (a) 2
   (b) 1
   (c) 2
   (d) 2
   (e) 2
   (f) 6
4. Probability and Bayesian Networks (15 points)
   (a) 5
   (b) 5
   (c) 5
5. Hidden Markov Models (15 points)
   (a) 5
   (b) 5
   (c) 5
6. Evolutionary Computing (15 points)
   (a) 2
   (b) 2
   (c) 2
   (d) 2
   (e) 2
   (f) 3
   (g) 2
7. Rough Sets (15 points)
   (a) 7
   (b) 8
7. Expert Systems (15 points)
   (a) 7
   (b) 8
8. Expert Systems (15 points)
   (a) 3
   (b) 3
   (c) 3
   (d) 3
   (e) 3
9. Logic Puzzle (15 points)

Total: 145 possible – graded out of 100.

Total:

Name:

Student Number:

1. Logic (15 points)

(a) If it is true that Bob goes to the store and false that Cheryl goes to work, then what are the truth-values of the following propositions?

(1) Bob goes to the store and Cheryl goes to work. F

(2) Either Bob goes to the store or Cheryl goes to work. T

(3) For Bob to go to the store it is sufficient that Cheryl not go to work. T

(b) If P is T and (P & Q) is F, then state the truth values of

(1) $P \vee Q$ T

(2) $-P \vee Q$ F

(3) $Q \rightarrow P$ T

(c) Show how one would represent the following natural language statements in predicate calculus form.

(1) Sidney loves Mary.

Loves (Sidney, Mary)

(2) If Fred is a dog and Bruce is a cat, then Fred chases Bruce.

[dog(Fred) & cat(Bruce)] → chases(Fred, Bruce)

(3) No one likes Bil.

~x [person(x) & likes(x, Bill)]

(4) All dogs chase some cat (or other).

(x) ∃y [dog(x) → cat(y) & chases(x,y)]

(5) All men are mortal.

(x) [man(x) → mortal(x)]

(6) Someone is loved by everybody.

∃y (x) [loves(x,y)]

(d) Using the following symbolizations:

Ax := "x is an aardvark"
Vx := "x is from the Transvaal"
Fx := "x is an ant"
Sx := "x likes to sleep in the sun"
Cx := "x is a cobra"
Dx := "x has a strange diet"
Mx := "x is a mongoose"
Exy := "x likes to eat y"

Translate the following formulas into sensible English:

(1) $\forall x\, ((Mx \,\&\, Vx) \rightarrow Dx)$

Any mongoose from the Transvaal has a strange diet

(2) $\exists x \exists y\, (Cx \,\&\, My \,\&\, Sx \,\&\, Sy)$

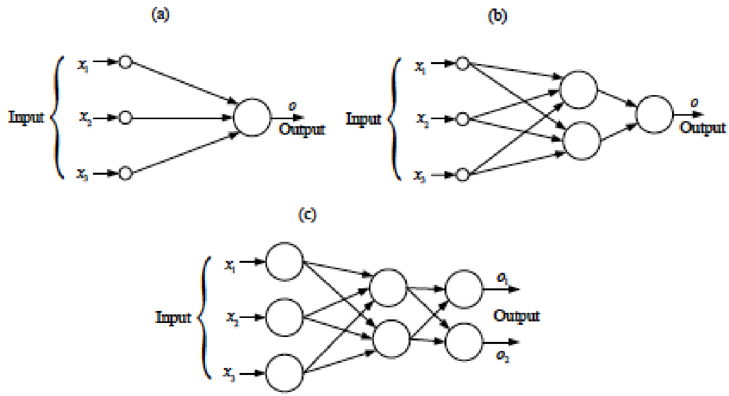
Some cobras and Mongooses like to sleep in the sun.

(3) $\forall x\, ((Mx \vee Ax) \rightarrow \forall y\, (Fy \rightarrow Exy))$

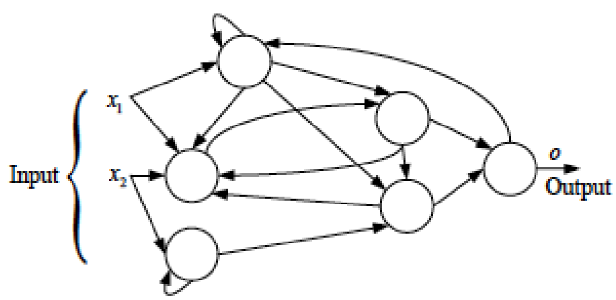
All mongooses and aardvarks like to eat ants

2. Neural Networks (15 points)

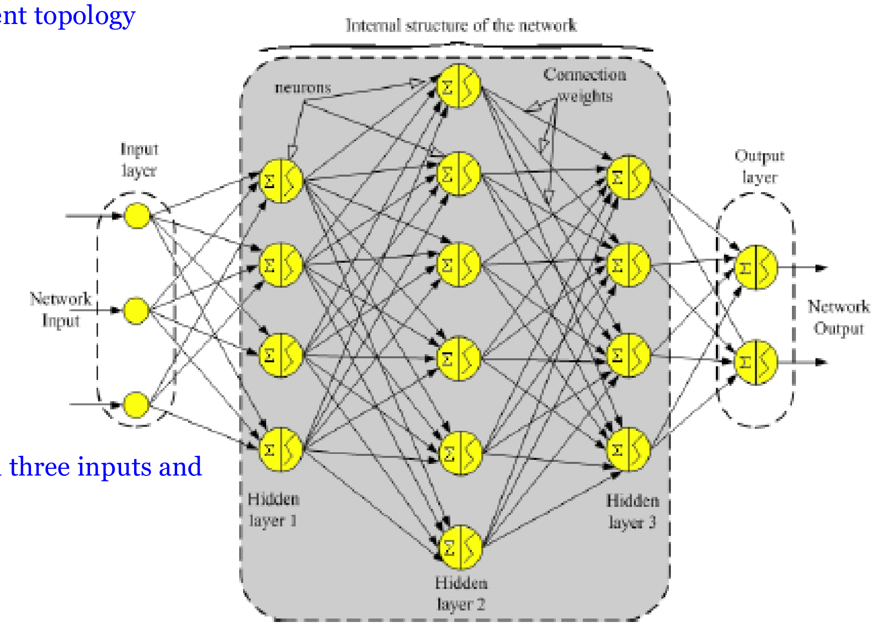
(a) Describe the characteristics of the following feed-forward neural networks:



(a) multi-input, single output, no hidden layer
(b) multi-input, single output, one hidden layer
(c) multi-input, multi-output, one hidden layer



A typical neural network with recurrent topology



Typical representation of a feed-forward (unidirectional) artificial neural network with three inputs and two outputs

(b) Representing boolean functions as neurons: In the case of arbitrary fan-in, and allowing for negations of variables (both are required), let $x_1, \ldots, x_N$ be N boolean variables, taking on the values 0 or 1.

(1) The conjunction, or AND, of these variables is defined by

$$x_1 \wedge \ldots \wedge x_N = 1 \text{ if and only if } x_i = 1 \text{ for all } i \tag{1}$$

Find weights w_i and threshold θ so that conjunction can be expressed as a neuron.

One possible solution:

$x_1 \wedge x_2 \wedge \ldots \wedge x_n = H(\sum_{i=1}^{n} xi - N + 0.5)$ where $w_i = 1$ and $\theta = N-0.5$.

(2) The disjunction, or OR, of the variables is defined by

$$x_1 \vee \ldots \vee x_N = 1 \text{ if and only if } x_i = 1 \text{ for some } i \tag{2}$$

Find weights w_i and threshold θ so that disjunction can be expressed as a neuron.

One possible solution:

$x_1 \vee x_2 \vee \ldots \vee x_n = H(\sum_{i=1}^{n} xi - N + 0.5)$ where $w_i = 1$ and $\theta = 0.5$.

(3) Consider the conjunction $x_1 \wedge x_2 \wedge \ldots \wedge x_n \wedge x_{n+1} \wedge \ldots \wedge x_N$ in which the first n variables are negated. Express this as a neuron.

One possible solution:

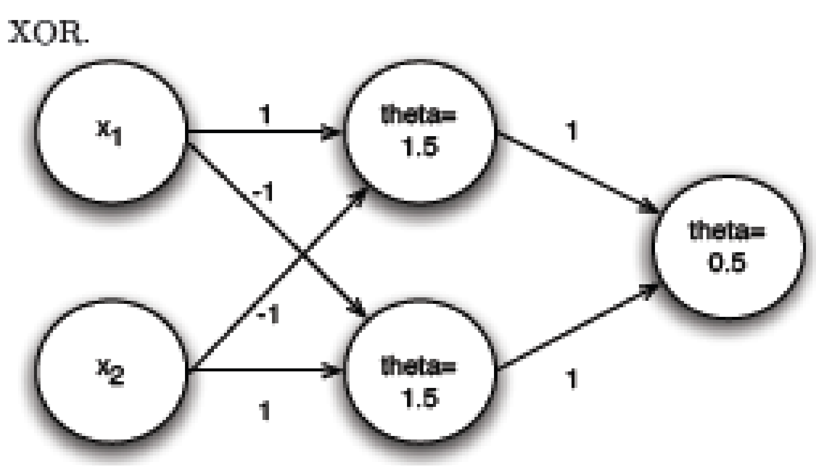
$x_1 \wedge x_2 \wedge \ldots \wedge x_n \wedge x_{n+1} \wedge \ldots \wedge x_N = H(\sum_{i=1}^{n} (1 - xi) + \sum_{i=n+1}^{N} xi - N + 0.5)$ where $w_i = -1$ for $i=1, 2, \ldots, n$ and $w_i=1$ for $i=n+1, n+2,/\ldots N$ and $\theta = 0.5 - n$.

(4) Do the same for the disjunction $x1 \vee x2 \vee \ldots \vee x_n \vee x_{n+1} \vee \ldots \vee x_N$.

One possible solution:

$x_1 \vee x_2 \vee \ldots \vee x_n \vee x_{n+1} \vee \ldots \vee x_N = H(\sum_{i=1}^{n} (1 - xi) + \sum_{i=n+1}^{N} xi - N - 0.5)$ where $w_i = -1$ for $i=1, 2, \ldots, n$ and $w_i=1$ for $i=n+1, n+2,/\ldots N$ and $\theta = 0.5 - n$.

(c) XOR as a 2-layer perceptron. We showed that a single neuron cannot compute the XOR function of two input variables. Construct a two-layer perceptron that computes XOR.



3. **Fuzzy Logic** (15 points)

(a) Is fuzzy logic a positive or a negative operation? Why?

Fuzzy logic is a positive operation because it is a way to gain conclusions from vague, ambiguous or imprecise information, although it requires a deep understanding of the system, and exact equations to achieve the correct answers. Fuzzy Logic allows expressing knowledge with subjective concepts such as very hot, bright red, and a long time, which are mapped into exact numeric ranges. Over the past years it has been gaining a lot of acceptance because is has a many benefits such as performance, simplicity, lower cost, and productivity.

(b) Fuzzy logic allows conclusion to be stated as:

(1) probabilities rather than certainties

(2) certainties and probabilitities

(3) probabilities rather than conclusions

(4) all the above

(c) Is fuzzy logic useful? Why?

Yes, it is very useful because it provides a simple way to draw many conclusions from imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. Unlike clasic logic, Fuzzy logic incorporates an alternative way of thinking and doesn't require a deep understanding of a system, exact equations, and precise numeric values. It is a powerful solving-problem methodology.

(d) A fuzzy set is fully defined by it membership functions. What is a membership function?

A membership function is a function in [0,1] that represents the degree of belonging. Formal defn in notes. Give at least one example

(e) Given the two fuzzy sets:

A = 0.3/1 + 0.6/2 + 0.45/3

B = 0.25/1 + 0.3/3 + 0.7/4

Calculate A & B union A & B intersection

Union = OR = MAX = 0.3/1 + 0.6/2 + 0.45/3 + 0.7/4

Intersection = AND = MIN = 0.25/1 + 0.3/3

(f) Compare and contrast a Knowledge based approach and a fuzzy logic approach to decision making

Knowledge based systems use the knowledge from an expert to make a decision. KBS include knowledge acquisition, inferencing, and knowledge representation.

Working memory: conclusions reached, data input by the user and details of system items, which have been checked.

Inferencing engine: deriving conclusions and control working of the system.

Production rules: IF then type of statements, conditions and conclusions.

Fuzzy Logic (FL) is used for modeling uncertainty and imprecision.

Fuzzy set: an object is ether a member (1) of a set or not (0), in fuzzy sets members have a degree of membership e.g. 0.8, 0.9, 0.2

Membership functions: function enables the determination the degree of membership.

A knowledge based system is based on crisp if/then condition/action rules, Fuzzy logic is based on a fuzzy rules.

4. Probability and Bayesian Networks (15 points)

Your Doritos are gone, and you have two apartment mates as suspects - Marc and Steve. You know the following things:

1. In previous chip-swiping incidents Marc was implicated 85% of the time, and Steve only 15% of the time.

2. Your across-the-hall neighbor believes she saw Marc eating your Doritos last night, but her eyesight is notoriously poor. You estimate that her rate of Marc/Steve differentiation is only 80% (i.e. 80% of the time she thinks she sees Marc it's actually Marc, and 20% of the time it's actually Steve. And if she thinks she sees Steve she's correct 80% of the time). Using Bayes rule:

(a) Given the above data what is the probability that Marc purloined the Doritos? Show your work.

WM means the witness reported Marc, DM means Doritoes were taken by Mac, DS means the Doritoes were taken by Steve. We are trying to determine p(DM|WM):

p(DM|WM) = p(WM|DM)p(DM) / p(WM) by Bayes' rules
= .85*.8 / p(WM,DM)+p(WM,DS)
= .68 / p(DM|WM)p(DM) + p(DS|WM)p(DS)
= .68 .85*.8+.2*.15
= .68 / .71 = .96

So there's a 96% chance that Marc took the Doritos.

(b) What would the probability be that Marc stole the chips if your neighbor thought she saw Steve? Show your work.

p(DM|WS) = p(WS|DM)p(DM) / p(WS)
= .2*.85 / p(DM|WS)p(DM) + p(DS|WS)p(DS)
= .2*.85 / .2*.85+.8*.15
= .17 / .17 + .12
= .58

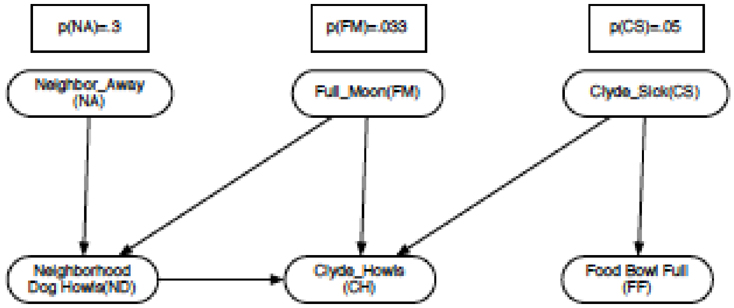
So even though our neighbor saw Steve, there's still a greater chance that Marc took the Doritoes.

Your loyal dog Clyde has been howling for the last three hours and you want to decide whether or not to take him to the vet or just to put in ear plugs and go back to sleep. You know that Clyde often howls when there's a full moon, when he's genuinely sick, or occasionally when a particular neighborhood dog starts howling. That neighborhood dog sometimes howls at the full moon and sometimes howls when her owner isn't home, but is not affected by Clyde's howls. If Clyde's really sick he probably won't have eaten very much and should have a bunch of food left in his bowl - but he sometimes just isn't very hungry despite not being sick.

(c) Create a Bayesian network for the scenario described above - use single letter names for each boolean variable (explaining what they mean of course). You can make up the exact numbers in the conditional probability tables (CPTs), but both the CPT values and the causal topology of the network should be reasonable. Briefly explain why you are setting up the topology as you are.

Here is one solution:

In terms of the CPTs, you just needed to make sure that adding multiple causes increased the probability of a thing - for instance, if the neighbor's dog is howling AND Clyde is sick it should be more probable that Clyde is howling than if only a single one of those causes was true. Additionally, you should have included something around 1/28 for the prior probability of Full Moon (assuming you're on Earth, which is a reasonable assumption).



FM	NA	p(ND)
T	T	.8
T	F	.4
F	T	.5
F	F	0

FM	CS	ND	p(CH)
T	T	T	.99
T	T	F	.90
T	F	T	.65
T	F	F	.4
F	T	T	.75
F	T	F	.5
F	F	T	.2
F	F	F	0

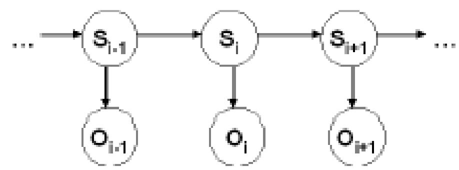
CS	p(FF)
T	.6
F	.1

5. Hidden Markov Models (15 points)

Hidden Markov Models are used for a variety of sequential data processing. As shown in the figure, the model includes a hidden layer, which described the data as a sequence of pre-defined states. The output layer maps to the observed signals that are emitted from the unknown states.

For example, in OCR systems, the hidden nodes represent the sequence of the underlying true characters. The observed decoded characters are not necessarily identical to the hand-written ones, due to the high variability of handwriting. For example, the hand-written “i” character is decoded as either “i”, “j”, or “l” with high probabilities. The advantage of using HMM for this problem is that it allows modeling context. For example, if the previous (hidden) state was assumed to be “q”, then the current character is more likely to be a vowel.

Given the observed OCR output sequence, the final output to the user is the sequence of states, which maximizes the joint probability of the HMM model.



(a) In the described OCR scenario, what is the size of the CPT table between two states of the hidden layer? What is the size of the emission probabilities table for every state?

There are as many states as the size of the alphabet. Thus, the transition table between two states is of size Size-of-alphabet°— Size-of-alphabet.

Given any state, there may be up to Size-of-alphabet non-zero entries in the emission table, representing the chance that the OCR recognizes the actual letter as any letter in the alphabet. That is, for every given state, the size of the emission probabilities table is Size-of-alphabet °— 1.

(b) Suggest a way for obtaining the transition and omission probabilities for the model.

In many cases, the relevant probabilities are obtained from manually labeled datasets. In the OCR scenario, this means that human annotators first tag a corpus of handwritten text with the correct machine-symbols (the relevant Ascii code, for example). Then, aligning a large annotated corpus with the corresponding OCR output can derive the emission probabilities.

As for the transition probabilities between states – processing machine-readable documents and counting the regularities of the transitions between consecutive letters can obtain these more cheaply.

(c) Suggest an HMM representation for the problem of speech recognition, or automatic Part of Speech labeling. (POS labeling is task of assigning every word its grammatical label, e.g., noun, verb, preposition etc.) You may suggest an HMM modeling for other problems as well.

Your suggestion should include the definition of states, the definition of the omitted signals, and a description of the values the hidden and observed nodes can take.

Explain why you think that the addressed problem would benefit using an HMM model.

In automatic Part-of-Speech (POS) labeling, the states are a pre-defined set of POS tags (e.g., noun, adjective etc.). The emitted signals are the actual words. This representation assumes that there are contextual regularities in the sequences of POS tags (for example, nouns may follow a determiner, but verbs may not.). While most words have only one possible POS tag that can be extracted from a dictionary (e.g., “from” is a Preposition in all cases), there are words with several possible labels (e.g., “walk” can be both a noun and a verb). Using an HMM allows to resolve such ambiguous cases.

HMMs are very dominant in the area of speech processing. There, a separate HMM model is built for every word. A vector of acoustic features is computed every 10 to 30 msec. The states of the HMM are therefore the phonetics of the word, and the observed signals are the acoustic signal representations. As for a human, mapping the measured signals to the correct phonetics is more precise if a sequence is considered rather than every signal in isolation.

6. Evolutionary Computing (15 points)

(a) Define a genetic algorithm

The genetic algorithm is a probabilistic search algorithm that iteratively transforms a set (called a population) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover (sexual recombination) and mutation.

(b) Why is probabilistic selection based on fitness?

- Better individuals are preferred
- Best is not always picked
- Worst is not necessarily excluded
- Nothing is guaranteed
- Mixture of greedy exploitation and adventurous exploration
- Similarities to simulated annealing (SA)

(c) If your search space consisted of an alphabet of size 2 and a chromosome (genome) of length 81 why would random or enumerative search be impractical?

Although 81-bit problems are very small for GA, however, even if *the length* is as small as 81, $2^{81} \sim 10^{27}$ = number of nanoseconds since the beginning of the universe 15 billion years ago

(d) Explain the mutation operation.

- Select 1 parent probabilistically based on fitness
- Pick point from 1 to NUMBER-OF-POINTS
- Delete sub tree at the picked point
- Grow new sub tree at the mutation point in same way as generated trees for initial random population (generation 0)
- The result is a syntactically valid executable program
- Put the offspring into the next generation of the population

(e) Explain the crossover operation.

- Select 2 parents probabilistically based on fitness
- Randomly pick a number from 1 to NUMBER-OF-POINTS for 1st parent
- Independently randomly pick a number for 2nd parent
- The result is a syntactically valid executable program
- Put the offspring into the next generation of the population
- Identify the sub trees rooted at the two picked points

(f) Give the general scheme of an Evolutionary Algorithm in pseudo-code.

```
BEGIN  INITIALIZE population with random candidate solutions;
       EVALUATE each candidate;
       REPEAT UNTIL (TERMINATION CONDITION is satisfied)       DO
              1.     SELECT parents;
              2.     RECOMBINE pairs of parents;
              3.     MUTATE the resulting offspring;
              4.     EVALUATE new candidate;
              5.     SELECT individuals for the next generation;    OD
END
```

(g) What are the components of an Evolutionary Algorithm?

- Representation (definition of individuals)
- Evaluation function (or fitness function)
- population
- parent selection mechanism
- variation operators, ecombination and mutation
- survivor selection mechanism (replacement)

7. Rough Sets (15 points)

(a) Given a set of objects, OBJ, a set of object attributes, AT, a set of values, VAL, and a function f:OBJ x AT -> VAL, so that each object is described by the values of its attributes, we define an equivalence relation R(A), where A is a subset of AT:

given two objects, o1 and o2,

$$o1\ R(A)\ o2 \Leftrightarrow f(o1,a) = f(o2, a), \forall a \text{ in } A$$

We say o1 and o2 are indiscernible (with respect to attributes in A). Now, we use this relation to partition the universe into equivalence classes, {e_0, e_1, e_2, ... , e_n} = R(A)*.

The pair (OBJ, R) form an "approximation space" with which we approximate arbitrary subsets of OBJ referred to as "concepts". Given O, an arbitrary subset of OBJ, we can approximate O by a union of equivalence classes:

the LOWER approximation of O (also known as the POSITIVE region):

$$\text{LOWER}(O) = \text{POS}(O) \Leftrightarrow \forall [o_R] \subseteq O\}$$

the UPPER approximation of O:

$$\text{UPPER}(O) \Leftrightarrow \forall [o_R] \cap O \neq \Theta$$

NEG(O) = OBJ - POS(O)
BND(O) = UPPER(O) - LOWER(O)   (boundary)

There are several versions of the exact definition of a rough set (unfortunately), the most common is that a roughly definable set is a set, O, such that BND(O) is non-empty. So a rough set is a set defined only by its lower and upper approximation. A set, O, whose boundary is empty is exactly definable.

If a subset of attributes, A, is sufficient to create a partition R(A)* which exactly defines O, then we say that A is a "reduct". The intersection of all reducts is known as the "core".

This is the simplest model. There are several probabilistic versions. Many researchers have used rough set theory for inductive learning systems, generating rules of the form:

description(POS(O)) → positive decision class
description(NEG(O)) → negative decision class
description(BND(O)) ~~> (probabilistically) positive decision class

Given the sample table below

Name	Education	Decision (Good Job Prospects)
Joe	High School	No
Mary	High School	Yes
Peter	Elementary	No
Paul	University	Yes
Cathy	Doctorate	Yes

So, the set of positive examples of people with good job prospects:

O = {Mary, Paul, Cathy}

The set of attributes:

A = AT = {Education}

The equivalence classes:

R(A)* = {{Joe, Mary}, {Peter}, {Paul}, {Cathy}}

The lower approximation and positive region:

POS(O) = LOWER(O) = {Paul, Cathy}

The negative region:

NEG(O) = {Peter}

The boundary region:

BND(O) = {Joe, Mary}

The upper approximation:

UPPER(O) = POS(O) + BND(O) = {Paul, Cathy, Joe, Mary}

As an aside, decision rules we can derive:

des(POS(O)) --> Yes

des(NEG(O)) --> No

des(BND(O)) ~~> Yes   (equivalently ~~> No)

That is:

(Education, University) or (Education, Doctorate) →Good prospects

(Education, Elementary) →No good prospects

(Education, High School) ~~> Good prospects (i.e., possibly)]

(b) Fill in the missing word/formula in the blank spaces below [hint: the last 4 are the same word].

Let $S = (U, A)$ be an information system, and $B \subseteq A$. A binary relation $IND_S(B)$ defined in the following way

$$IND_S(B) = \{(x, x') \in U^2 \mid \forall a \in B \; a(x) = a(x')\}$$

is called the *B-indiscernibility relation*. It is easy to see that $IND_S(B)$ is equivalence relation. If $(x\, x') \in IND_S(B)$, then objects x and x' are indiscernible from each other by attributes from B. The equivalence classes of the *B-indiscernibility* relation are denoted $[x]_B$. The subscript S in the indiscernibility relation is usually omitted if it is clear which information system is meant.

Let $S=(U,A)$ be an information system, $B \subseteq A$, and let $a \in B$.

We say that a is *dispensable* in B if $IND_S(B) = IND_S(B \{a\})$; otherwise a is *indispensable* in B.

A set B is called *independent* if all its attributes are indispensable.

Any subset B' of B is called a *reduct* of B if B' is independent and $IND_S(B') = IND_S(B)$.

Hence, a reduct is a set of attributes that preserves partition. It means that a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whol set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe.

8. Expert Systems (15 points)

(a) What is an expert system?

Expert Systems are computer programs that exhibit intelligent behavior. They are concerned with the concepts and methods of symbolic inference, or reasoning, by a computer, and how the knowledge used to make those inferences will be represented.

Achieving expert-level competence in solving problems in task areas by bringing to bear a body of knowledge about specific tasks is called *knowledge-based* or *expert system*. The term expert system is reserved for programs whose knowledge base contains the knowledge used by human experts. Expert systems and knowledge-based systems are used synonymously. The area of human intellectual endeavor to be captured in an expert system is called the *task domain*. *Task* refers to some goal-oriented, problem-solving activity. *Domain* refers to the area within which the task is being performed. Typical tasks are diagnosis, planning, scheduling, configuration and design.

(b) What are the components of Expert Systems?

knowledge base
- contains essential information about the problem domain
- often represented as *facts* and *rules*

inference engine
- mechanism to derive new knowledge from the knowledge base and the information provided by the user
- often based on the *use of rules*

user interface
- interaction with end users
- development and maintenance of the knowledge base

(c) What is a production rule and how does it work?

A *production rule* consists of an IF part and a THEN part (also called a *condition* and an *action*). The IF part lists a set of conditions in some logical combination. The piece of knowledge represented by the production rule is relevant to the line of reasoning being developed if the IF part of the rule is satisfied; consequently, the THEN part can be concluded, or its problem-solving action taken. Expert systems whose knowledge is represented in rule form are called *rule-based systems*.

(d) What is the difference between forward chaining and backward chaining?

Forward chaining and backward chaining are different methods of reasoning and rule ctivation

- forward chaining (data-driven)
    - reasoning from facts to the conclusion
    - as soon as facts are available, they are used to match antecedents of rules
    - a rule can be activated if all parts of the antecedent are satisfied
    - often used for real-time expert systems in monitoring and control
    - examples: CLIPS, OPS5
- backward chaining (query-driven)
    - starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
    - often used in diagnostic and consultation systems
    - examples: EMYCIN

(e) What are some advantages of using expert systems and what are some of the problems?

- Advantages
    - Economical - lower cost per user
    - Availability - accessible anytime, almost anywhere
    - response time - often faster than human experts

- reliability - can be greater than that of human experts; no distraction, fatigue, emotional involvement, ...
- explanation - reasoning steps that lead to a particular conclusion
- intellectual property - can't walk out of the door

- Problems
  - limited knowledge
    - "shallow" knowledge - no "deep" understanding of the concepts and their relationships
    - no "common-sense" knowledge
    - no knowledge from possibly relevant related domains
    - "closed world"
      - the XPS knows only what it has been explicitly "told"
      - it doesn't know what it doesn't know
  - mechanical reasoning
    - may not have or select the most appropriate method for a particular problem
    - some "easy" problems are computationally very expensive
  - lack of trust
    - users may not want to leave critical decisions to machines

9. Logic Puzzle (15 points)

Happy Birthday: Four women each sent one birthday card last week to a friend who lives in a city abroad. When did each woman post her card, what is the name of her friend, and in which city does each friend live?

1. The card to Aileen was posted three days later than the one to Johannesburg.
2. Madge posted her card later in the week than the one to Rome, but earlier in the week than the one sent by Muriel to Amy, who doesn't live in Tokyo.
3. Aster's card was sent by Miriam, but not to an address in Rome.

	Day				Friend				City			
	Mon	Tues	Thurs	Fri	Aileen	Alison	Amy	Aster	Johannesburg	Paris	Rome	Tokyo
Madge												
Megan												
Miriam												
Murial												
Johhan-esburg												
Paris												
Rome												
Tokyo												
Aileen												
Alison												
Amy												
Aster												

Sender	Day	Friend	City
Madge	Thursday	Aileen	Tokyo
Megan	Tuesday	Alison	Rome
Miriam	Monday	Aster	Johannesburg
Muiel	Friday	Amy	Paris

The card sent to Amy, who doesn't live in Rome or Tokyo (clue 2) was posted on either Thursday or Friday. The card sent to Johannesburg was posted on either Monday or Tuesday (clue 1), so wasn't addressed to Amy. Thus Amy lives in Paris. Aileen's card was also posted on either Thursday or Friday (1) and the card sent to Rome was posted on either Monday or Tuesday (2), so (by elimination) Aileen lives I n Tokyo. Miriam sent a card to Aster (3) who doesn't live in Rome, so Johannesburg. Alison lives in Rome; her card wasn't sent by Madge (2) and Madge didn't post any card to Amy. So Madge sent a card to Aileen. Madge's card wasn't posted on Friday (2) so on Thursday (1) and the one to Johannesburg was posted on Monday. The card to Rome was posted on Tuesday (2) so (by elimination) Muriel posed a card on Friday, and Megan on Tuesday.