Sample Paper
MCA IV (Sem)
Artificial Intelligence (MCA-404)
Session 2018-19

Time : 3 Hrs **Maximum Marks : 80**

Minimum Passing Marks : 32

Attempt all questions. Marks of questions are indicated against each question.

1 Answer each part in one line

(a) What is intelligence?

Intelligence is the ability to the ability to acquire and apply knowledge and skills.

(b) What is state space?

State space search is a process in which successive configurations or states of an instance are considered, with the intention of finding a goal state with a desired property.

(c) Define fuzziness?

Fuzziness is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1 inclusive.

(d) What is frame?

A frame is an artificial intelligence data structure used to divide knowledge into substructures by representing "stereotyped situations."

(e) What are quantifiers?

In logic, quantification specifies the quantity of specimens in the domain of discourse that satisfy an open formula. The two most common quantifiers mean "for all" and "there exists". The traditional symbol for the universal quantifier "all" is "∀", a rotated letter "A", and for the existential quantifier "exists" is "∃", a rotated letter "E".

(f) Define heuristic.

A heuristic function, also called simply a heuristic, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow.

(g) Define Rote Learning.

Rote learning also known as Cache Learning is the basic learning activity. It is also called memorization because the knowledge, without any modification is, simply copied into the knowledge base. As computed values are stored, this technique can save a significant amount of time.

(h) What is MYCIN?

MYCIN was an early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient's body weight — the name derived from the antibiotics themselves

(i) What is LISP?

LISP, an acronym for list processing, is a programming language that was designed for easy manipulation of data strings. Developed in 1959 by John McCarthy, it is a commonly used language for artificial intelligence (AI) programming.

(j) What is semantic network?

A semantic network is a graphic notation for representing knowledge in patterns of interconnected nodes. Semantic networks became popular in artificial intelligence and natural language processing only because it represents knowledge or supports reasoning.

2 **Answer the following questions in not more than 50 words.**

(a) Define Heuristic Search.

A Heuristic is a technique to solve a problem faster than classic methods, or to find an approximate solution when classic methods cannot. This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed. A Heuristic (or a heuristic function) takes a look at search algorithms. At each branching step, it evaluates the available information and makes a decision on which branch to follow. It does so by ranking alternatives. The Heuristic is any device that is often effective but will not guarantee work in every case.

(b) Define predicate logic. How following can be represented in predicate logic?

1) All dogs have tails.

First-order logic is symbolized reasoning in which each sentence, or statement, is broken down into a subject and a predicate. The predicate modifies or defines the properties of the subject. In first-order logic, a predicate can only refer to a single subject. First-order logic is also known as first-order predicate calculus or first-order functional calculus.

A sentence in first-order logic is written in the form Px or P(x), where P is the predicate and x is the subject, represented as a variable. Complete sentences are logically combined and manipulated according to the same rules as those used in Boolean algebra.

$\forall x : \text{Dog}(x) \text{ AND Tail}(x)$

(c) Describe Monotonic reasoning.

Monotonic learning is when an agent may not learn the knowledge that contradicts with what it already known or exists, it will not replace a statement with its negation. Thus, the knowledge base may only grow with new facts in a monotonic fashion. The advantages of monotonic learning are:

1) Greatly simplified truth-maintenance
2) Greater choice in learning strategies

(d) Define Back Tracking.

Backtracking is an algorithm for capturing some or all solutions to given computational issues, especially for constraint satisfaction issues. The algorithm can only be used for problems which can accept the concept of a “partial candidate solution” and allows a quick test to see if the candidate solution can be a complete solution. Backtracking is considered an important technique to solve constraint satisfaction issues and puzzles. It is also considered a great technique for parsing and also forms the basis of many logic programming languages.

(e) Explain Hill Climbing.

Hill Climbing is heuristic search used for mathematical optimization problems in the field of Artificial Intelligence .

Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem. This solution may not be the global optimal maximum.

In the above definition, mathematical optimization problems implies that hill climbing solves the problems where we need to maximize or minimize a given real function by choosing values from the given inputs. Example-Travelling salesman problem where we need to minimize the distance traveled by salesman.

'Heuristic search' means that this search algorithm may not find the optimal solution to the problem. However, it will give a good solution in reasonable time.

3 Answer the following questions in not more than 150 words.

(a) Explain Dempster- Shafer theory.

The Dempster-Shafer theory, also known as the theory of belief functions, is a generalization of the Bayesian theory of subjective probability. Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question on probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ from probabilities will depend on how closely the two questions are related.

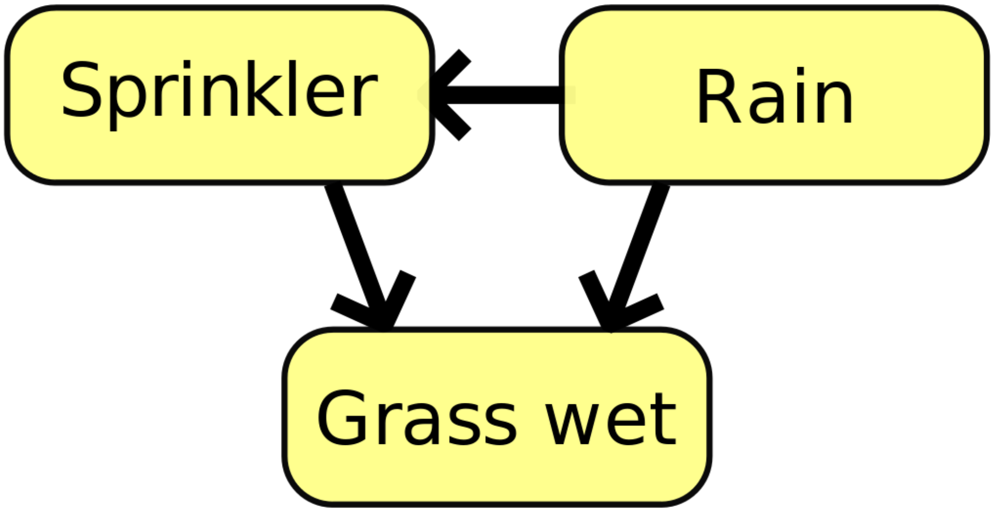
The Dempster-Shafer theory is based on two ideas: the idea of obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule for combining such degrees of belief when they are based on independent items of evidence.

To illustrate the idea of obtaining degrees of belief for one question from subjective probabilities for another, suppose I have subjective probabilities for the reliability of my friend Betty. My probability that she is reliable is 0.9, and my probability that she is unreliable is 0.1. Suppose she tells me a limb fell on my car. This statement, which must true if she is reliable, is not necessarily false if she is unreliable. So her testimony alone justifies a 0.9 degree of belief that a limb fell on my car, but only a zero degree of belief (not a 0.1 degree of belief) that no limb fell on my car. This zero does not mean that I am sure that no limb fell on my car, as a zero probability would; it merely means that Betty's testimony gives me no reason to believe that no limb fell on my car. The 0.9 and the zero together constitute a belief function.

(b) Describe properties of Bayesian Network.

A Bayesian network, Bayes network, belief network, decision network, Bayes(ian) model or probabilistic directed acyclic graphical model is a probabilistic graphical model (a type of statistical model) that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Efficient algorithms can perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g. speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.



(c) State water jug problem with solution and production rules.

Statement :- We are given 2 jugs, a 4 liter one and a 3- liter one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can we get exactly 2 liters of water in to the 4-liter jugs?

Solution:-

The state space for this problem can be defined as

{ (i ,j) i = 0,1,2,3,4 j = 0,1,2,3}

'i' represents the number of liters of water in the 4-liter jug and 'j' represents the number of liters of water in the 3-liter jug. The initial state is (0,0) that is no water on each jug. The goal state is to get (2,n) for any value of 'n'.

To solve this we have to make some assumptions not mentioned in the problem. They are

1. We can fill a jug from the pump.
2. we can pour water out of a jug to the ground.
3. We can pour water from one jug to another.
4. There is no measuring device available.

(d) Describe Augmented Transition Net.

An augmented transition network (ATN) is a type of graph structure used in the operational definition of formal languages, used especially in parsing relatively complex natural languages, and having wide application in artificial intelligence. An ATN can, theoretically, analyze the structure of any sentence, however complicated.

ATNs build on the idea of using finite state machines to parse sentences. ATNs have states to mark the progress in processing a string. Transitions correspond to individual words or phrases from a syntactic type. W. A. Woods in "Transition Network Grammars for Natural Language Analysis" claims that by adding a recursive mechanism to a finite state model, parsing can be achieved much more efficiently. Instead of building an automaton for a particular sentence, a collection of transition graphs are built. A grammatically correct sentence is parsed by reaching a final state in any state graph. Transitions between these graphs are simply subroutine calls from one state to any initial state on any graph in the network. A sentence is determined to be grammatically correct if a final state is reached by the last word in the sentence.

(e) Describe Generate and Test algorithm.

GENERATE-AND-TEST ALGORITHM

Generate-and-test search algorithm is a very simple algorithm that guarantees to find a solution if done systematically and there exists a solution.

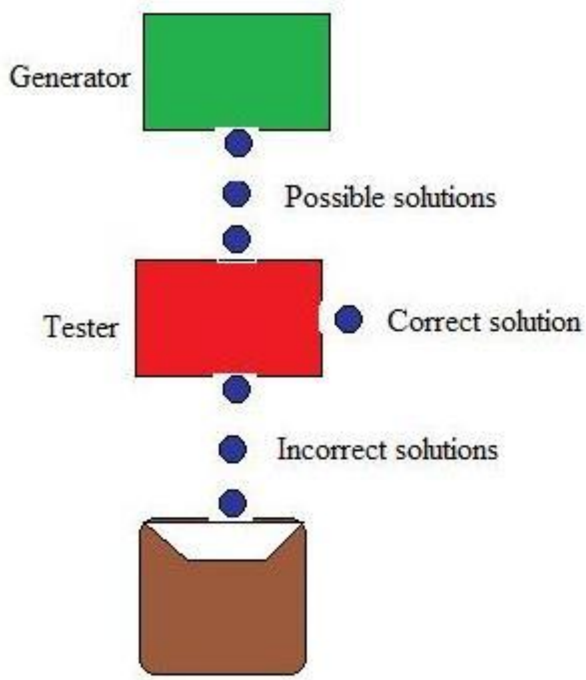
ALGORITHM: GENERATE-AND-TEST

1.Generate a possible solution.

2.Test to see if this is the expected solution.

3.If the solution has been found quit else go to step 1.

Potential solutions that need to be generated vary depending on the kinds of problems. For some problems the possible solutions may be particular points in the problem space and for some problems, paths from the start state.



Generate-and-test, like depth-first search, requires that complete solutions be generated for testing. In its most systematic form, it is only an exhaustive search of the problem space. Solutions can also be generated randomly but solution is not guaranteed. This approach is what is known as British Museum algorithm: finding an object in the British Museum by wandering randomly.

4 Answer following in detail

(a) Define Machine Learning.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Some machine learning methods
Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiringunlabeled data generally doesn't require additional resources.
Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

(b) Describe hill climbing search technique.

Hill Climbing is heuristic search used for mathematical optimization problems in the field of Artificial Intelligence .
Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem. This solution may not be the global optimal maximum.

In the above definition, mathematical optimization problems implies that hill climbing solves the problems where we need to maximize or minimize a given real function by choosing values from the given inputs. Example-Travelling salesman problem where we need to minimize the distance traveled by salesman.
'Heuristic search' means that this search algorithm may not find the optimal solution to the problem. However, it will give a good solution in reasonable time.

A heuristic function is a function that will rank all the possible alternatives at any branching step in search algorithm based on the available information. It helps the algorithm to select the best route out of possible routes.
Features of Hill Climbing

Variant of generate and test algorithm : It is a variant of generate and test algorithm. The generate and test algorithm is as follows :
1. Generate a possible solutions.
2. Test to see if this is the expected solution.
3. If the solution has been found quit else go to step 1.

Hence we call Hill climbing as a variant of generate and test algorithm as it takes the feedback from test procedure. Then this feedback is utilized by the generator in deciding the next move in search space.

Uses the Greedy approach : At any point in state space, the search moves in that direction only which optimizes the cost of function with the hope of finding the optimal solution at the end.
Types of Hill Climbing
Simple Hill climbing : It examines the neighboring nodes one by one and selects the first neighboring node which optimizes the current cost as next node.
Algorithm for Simple Hill climbing :
Step 1 : Evaluate the initial state. If it is a goal state then stop and return success. Otherwise, make initial state as current state.

Step 2 : Loop until the solution state is found or there are no new operators present which can be applied to current state.

a) Select a state that has not been yet applied to the current state and apply it to produce a new state.

b) Perform these to evaluate new state
i. If the current state is a goal state, then stop and return success.
ii. If it is better than the current state, then make it current state and proceed further.
iii. If it is not better than the current state, then continue in the loop until a solution is found.

Step 3 : Exit.

Steepest-Ascent Hill climbing : It first examines all the neighboring nodes and then selects the node closest to the solution state as next node.
Step 1 : Evaluate the initial state. If it is goal state then exit else make the current state as initial state
Step 2 : Repeat these steps until a solution is found or current state does not change

i. Let ‘target’ be a state such that any successor of the current state will be better than it;

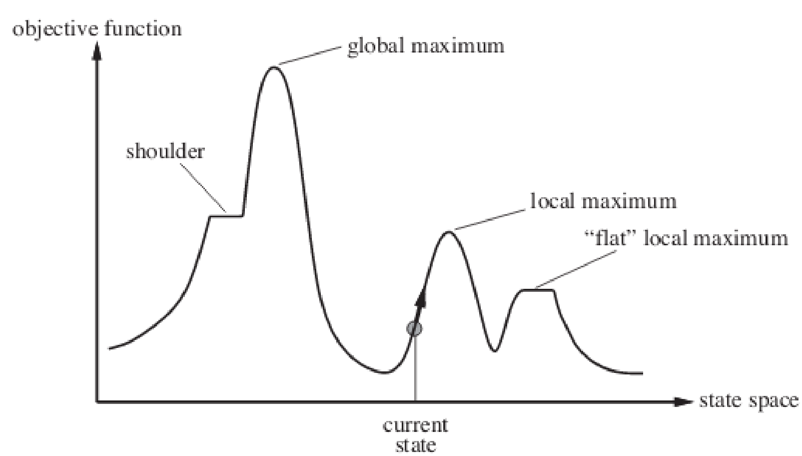
ii. for each operator that applies to the current state
   a. apply the new operator and create a new state
   b. evaluate the new state
   c. if this state is goal state then quit else compare with ‘target’
   d. if this state is better than ‘target’, set this state as ‘target’
   e. if target is better than current state set current state to Target

Step 3 : Exit

Stochastic hill climbing : It does not examine all the neighboring nodes before deciding which node to select .It just selects a neighboring node at random, and decides (based on the amount of improvement in that neighbor) whether to move to that neighbor or to examine another.
State Space diagram for Hill Climbing

State space diagram is a graphical representation of the set of states our search algorithm can reach vs the value of our objective function(the function which we wish to maximize).
X-axis : denotes the state space ie states or configuration our algorithm may reach.
Y-axis : denotes the values of objective function corresponding to to a particular state.
The best solution will be that state space where objective function has maximum value(global maximum).

State Space diagram for Hill climbing



Different regions in the State Space Diagram

Local maximum : It is a state which is better than its neighboring state however there exists a state which is better than it(global maximum). This state is better because here value of objective function is higher than its neighbors.

Global maximum : It is the best possible state in the state space diagram. This because at this state, objective function has highest value.

Plateua/flat local maximum : It is a flat region of state space where neighboring states have the same value.

Ridge : It is region which is higher than its neighbours but itself has a slope. It is a special kind of local maximum.

Current state : The region of state space diagram where we are currently present during the search.

Shoulder : It is a plateau that has an uphill edge.

Problems in different regions in Hill climbing

Hill climbing cannot reach the optimal/best state(global maximum) if it enters any of the following regions :

Local maximum : At a local maximum all neighboring states have a values which is worse than than the current state. Since hill climbing uses greedy approach, it will not move to the worse state and terminate itself. The process will end even though a better solution may exist.

To overcome local maximum problem : Utilize backtracking technique. Maintain a list of visited states. If the search reaches an undesirable state, it can backtrack to the previous configuration and explore a new path.

Plateau : On plateau all neighbors have same value . Hence, it is not possible to select the best direction.

To overcome plateaus : Make a big jump. Randomly select a state far away from current state. Chances are that we will land at a non-plateau region

Ridge : Any point on a ridge can look like peak because movement in all possible directions is downward. Hence the algorithm stops when it reaches this state.

To overcome Ridge : In this kind of obstacle, use two or more rules before testing. It implies moving in several directions at once.

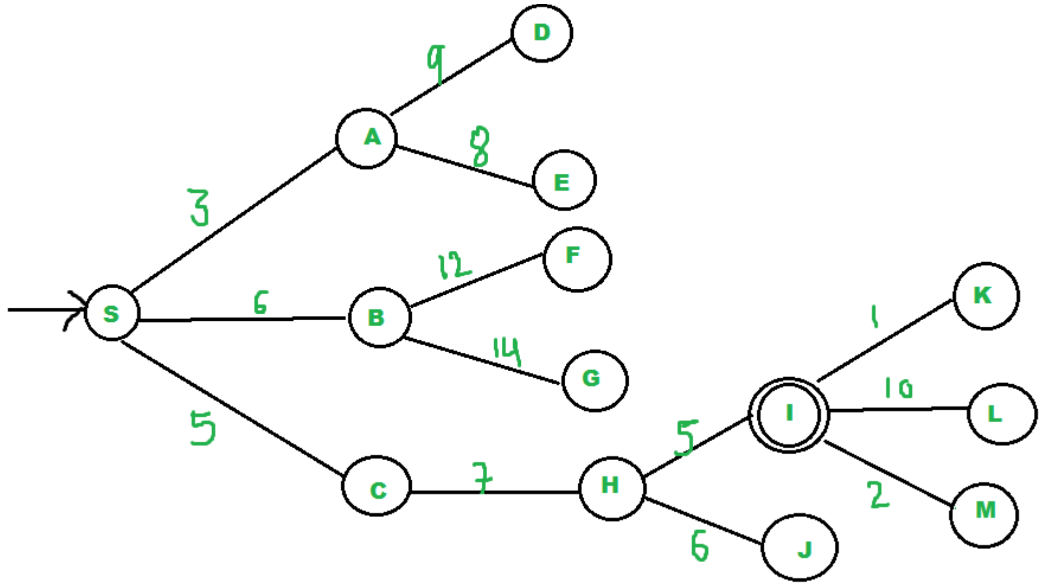
5 Explain following

(a) Best First Search

In BFS and DFS, when we are at a node, we can consider any of the adjacent as next node. So both BFS and DFS blindly explore paths without considering any cost function. The idea of Best First Search is to use an evaluation function to decide which adjacent is most promising and then explore. Best First Search falls under the category of Heuristic Search or Informed Search.

We use a priority queue to store costs of nodes. So the implementation is a variation of BFS, we just need to change Queue to PriorityQueue.

Let us consider below example.



Analysis :

The worst case time complexity for Best First Search is O(n * Log n) where n is number of nodes. In worst case, we may have to visit all nodes before we reach goal. Note that priority queue is implemented using Min(or Max) Heap, and insert and remove operations take O(log n) time.
Performance of the algorithm depends on how well the cost or evaluation function is designed.

(b) Features of frames
Frames were proposed by Marvin Minsky in his 1974 article "A Framework for Representing Knowledge." A frame is an artificial intelligence data structure used to divide knowledge into substructures by representing "stereotyped situations." Frames are the primary data structure used in artificial intelligence frame language.

Frames are also an extensive part of knowledge representation and reasoning schemes. Frames were originally derived from semantic networks and are therefore part of structure based knowledge representations. According to Russell and Norvig's "Artificial Intelligence, A Modern Approach," structural representations assemble "...facts about particular object and event types and arrange the types into a large taxonomic hierarchy analogous to a biological taxonomy."
The frame contains information on how to use the frame, what to expect next, and what to do when these expectations are not met. Some information in the frame is generally unchanged while other information, stored in "terminals," usually change. Terminals can be considered as variables. Top level frames carries information, that is always true about the problem in hand, however, terminals do not have to be true. Their value might change with the new information encountered. Different frames may share the same terminals.

Each piece of information about a particular frame is held in a slot. The information can contain:

Facts or Data
Values (called facets)
Procedures (also called procedural attachments)
IF-NEEDED : deferred evaluation
IF-ADDED : updates linked information
Default Values
For Data
For Procedures
Other Frames or Subframes

OR

(a) Minimal List Reasoning

An analogy is a comparison between two objects, or systems of objects, that highlights respects in which they are thought to be similar. Analogical reasoning is any type of thinking that relies upon an analogy. An analogical argument is an explicit representation of a form of analogical reasoning that cites accepted similarities between two systems to support the conclusion that some further similarity exists. In general (but not always), such arguments belong in the category of ampliative reasoning, since their conclusions do not follow with certainty but are only supported with varying degrees of strength. However, the proper characterization of analogical arguments is subject to debate (see §2.2).

Analogical reasoning is fundamental to human thought and, arguably, to some nonhuman animals as well. Historically, analogical reasoning has played an important, but sometimes mysterious, role in a wide range of problem-solving contexts. The explicit use of analogical arguments, since antiquity, has been a distinctive feature of scientific, philosophical and legal reasoning. This article focuses primarily on the nature, evaluation and justification of analogical arguments. Related topics include metaphor, models in science, and precedent and analogy in legal reasoning.

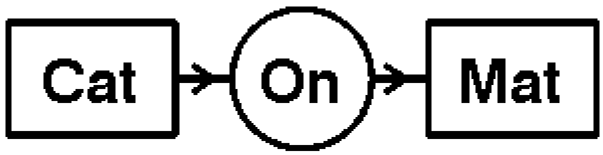
Analogies have a related (and not entirely separable) justificatory role. This role is most obvious where an analogical argument is explicitly offered in support of some conclusion. The intended degree of support for the conclusion can vary considerably. At one extreme, these arguments can be strongly predictive. For example (Example 1), hydrodynamic analogies exploit mathematical similarities between the equations governing ideal fluid flow and torsional problems. To predict stresses in a planned structure, one can construct a fluid model, i.e., a system of pipes through which water passes. Within the limits of idealization, such analogies allow us to make demonstrative inferences, for example, from a measured quantity in the fluid model to the analogous value in the torsional problem. In practice, there are numerous complications

(b) Conceptual Graph with example

Conceptual graphs are formally defined in an abstract syntax that is independent of any notation, but the formalism can be represented in several different concrete notations. This document illustrates CGs by means of examples represented in the graphical display form (DF), the formally defined conceptual graph interchange form (CGIF), and the compact, but readable linear form (LF). Every CG is represented in each of these three forms and is translated to a logically equivalent representation in predicate calculus and in the Knowledge Interchange Format (KIF). For the formal definition of conceptual graphs and the various notations for representing them, see the draft proposed American National Standard. For examples of an English-like notation for representing logic, see the web page on controlled English.

1. A cat is on a mat.

In the display form (DF), concepts are represented by rectangles: the concept [Cat] represents a instance of a cat, and [Mat] represents an instance of a mat. Conceptual relations are represented by circles or ovals: the conceptual relation (On) relates a cat to a mat. The arcs that link the relations to the concepts are represented by arrows: the first arc has an arrow pointing toward the relation, and the second arc has an arrow pointing away from the relation. If a relation has more than two arcs, the arcs are numbered.



In the linear form (LF), concepts are represented by square brackets instead of boxes, and the conceptual relations are represented by parentheses instead of circles:

[Cat]®(On)®[Mat].

Both DF and LF are designed for communication with humans or between humans and machines. For communication between machines, the conceptual graph interchange form (CGIF) has a syntax that uses co-reference labels to represent the arcs:

[Cat: *x] [Mat: *y] (On ?x ?y)

The symbols *x and *y are called defining labels. The matching symbols ?x and ?y are the bound labels that indicate references to the same instance of a cat x or a mat y. To reduce the number of coreference labels, CGIF also permits concepts to be nested inside the relation nodes:

(On [Cat] [Mat])