Dr. SNS RAJALAKSHMI COLLEGE OF ARTS & SCIENCE

Coimbatore – 49.

DEPARTMENT OF COMPUTER APPLICATIONS

LECTURE NOTES

CLASS	:	**I MCA**	**BATCH :**	**2023 - 2025**
COURSE	:	**INTRODUCTION TO ARTIFICIAL INTELLIGENCE**		
COURSE CODE	:	**21PCA101**	**SEMESTER :**	**I**

UNIT – I

- **What is AI?**

  Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

- **How does AI work?**
  - As the hype around AI has accelerated, vendors have been scrambling to promote how their products and services use AI. Often what they refer to as AI is simply one component of AI, such as machine learning.
  - AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular.
  - AI programming focuses on three cognitive skills
    - ✓ learning
    - ✓ reasoning
    - ✓ self-correction.
- **Learning processes.** This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.
- **Reasoning processes.** This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

- **Self-correction processes.** This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.
- Eden to ENIAC: Attitudes toward Intelligence, Knowledge, and Human Artifice:
  - This intelligence forms the foundation for all of human technology and ultimately all human civilization. The work of Aeschylus, the classical Greek dramatist, illustrates a deep and ancient awareness of the extraordinary power of knowledge. Artificial intelligence, in its very direct concern for Prometheus's gift, has been applied to all the areas of his legacy—medicine, psychology, biology, astronomy, geology—and many areas of scientific endeavor that Aeschylus could not have imagined.
  - Though Prometheus's action freed humanity from the sickness of ignorance, it also earned him the wrath of Zeus. Outraged over this theft of knowledge that previously belonged only to the gods of Olympus, Zeus commanded that Prometheus be chained to a barren rock to suffer the ravages of the elements for eternity
  - . The notion that human efforts to gain knowledge constitute a transgression against the laws of God or nature is deeply ingrained in Western thought. It is the basis of the story of Eden and appears in the work of Dante and Milton. Both Shakespeare and the ancient Greek tragedians portrayed intellectual ambition as the cause of disaster.
  - The belief that the desire for knowledge must ultimately lead to disaster has persisted throughout history, enduring the Renaissance, the Age of Enlightenment, and even the scientific and philosophical advances of the nineteenth and twentieth centuries. Thus, we should not be surprised that artificial intelligence inspires so much controversy in both academic and popular circles.
  - Many and long were the conversations between Lord Byron and Shelley to which I was a devout and silent listener. During one of these, various philosophical doctrines were discussed, and among others the nature of the principle of life, and whether there was any probability of its ever being discovered and communicated.

A Brief History of the Foundations for AI

- The logical starting point for such a history is the genius of Aristotle, or as Dante in the Divine Comedy refers to him, “the master of them that know”. Aristotle wove together the insights, wonders, and fears of the early Greek tradition with the careful analysis and disciplined thought that were to become the standard for more modern science.
- Aristotle, the most fascinating aspect of nature was change. In his Physics, he defined his “philosophy of nature” as the “study of things that change”. He distinguished between the matter and form of things: a sculpture is fashioned from the material bronze and has the form of a human. Change occurs when the bronze is molded to a new form. The matter/form distinction provides a philosophical basis for modern notions such as symbolic computing and data abstraction. In computing (even with numbers) we are manipulating patterns that are the forms of electromagnetic material, with the changes of form of this material representing aspects of the solution process.
- Abstracting the form from the medium of its representation not only allows these forms to be manipulated computationally but also provides the promise of a theory of data structures, the heart of modern computer science. It also supports the creation of an “artificial” intelligence.
- In his Metaphysics, beginning with the words “All men by nature desire to know”, Aristotle developed a science of things that never change, including his cosmology and theology. More relevant to artificial intelligence, however, was Aristotle’s epistemology or analysis of how humans “know” their world, discussed in his Logic. Aristotle referred to logic as the “instrument” (organon), because he felt that the study of thought itself was at the basis of all knowledge. In his Logic, he investigated whether certain propositions can be said to be “true” because they are related to other things that are known to be “true”.
- Thus, if we know that “all men are mortal” and that “Socrates is a man”, then we can conclude that “Socrates is mortal”. This argument is an example of what Aristotle referred to as a syllogism using the deductive form modus ponens. Although the formal axiomatization of reasoning needed another two thousand years for its full flowering in the works of Gottlob Frege, Bertrand Russell, Kurt Gödel, Alan Turing, Alfred Tarski, and others, its roots may be traced to Aristotle.
- There are two consequences of this analysis essential to the AI enterprise:

1. By attempting to separate the mind from the physical world, Descartes and related thinkers established that the structure of ideas about the world was not necessarily the same as the structure of their subject matter. This underlies the methodology of AI, along with the fields of epistemology, psychology, much of higher mathematics, and most of modern literature: mental processes have an existence of their own, obey their own laws, and can be studied in and of themselves.

2. Once the mind and the body are separated, philosophers found it necessary to find a way to reconnect the two, because interaction between Descartes mental, res cogitans, and physical, res extensa, is essential for human existence.

- Although millions of words have been written on this mind–body problem, and numerous solutions proposed, no one has successfully explained the obvious interactions between mental states and physical actions while affirming a fundamental difference between them. The most widely accepted response to this problem, and the one that provides an essential foundation for the study of AI, holds that the mind and the body are not fundamentally different entities at all. On this view, mental processes are indeed achieved by physical systems such as brains (or computers). Mental processes, like physical processes, can ultimately be characterized through formal mathematics. Or, as acknowledged in his Leviathan by the 17th century English philosopher Thomas Hobbes (1651), “By ratiocination, I mean computation”.

AI and the Rationalist and Empiricist Traditions

- Modern research issues in artificial intelligence, as in other scientific disciplines, are formed and evolve through a combination of historical, social, and cultural pressures. Two of the most prominent pressures for the evolution of AI are the empiricist and rationalist traditions in philosophy.
- The rationalist tradition, as seen in the previous section, had an early proponent in Plato, and was continued on through the writings of Pascal, Descartes, and Liebniz. For the rationalist, the external world is reconstructed through the clear and distinct ideas of a mathematics. A criticism of this dualistic approach is the forced disengagement of representational systems from their field of reference. The issue is whether the meaning attributed to a representation can be defined independent of its application conditions.

- Many AI programs have very much of this rationalist flavor. Early robot planners, for example, would describe their application domain or “world” as sets of predicate calculus statements and then a “plan” for action would be created through proving theorems about this “world” (Fikes et al. 1972, see also Section 8.4). Newell and Simon’s Physical Symbol System Hypothesis (Introduction to Part II and Chapter 16) is seen by many as the archetype of this approach in modern AI. Several critics have commented on this rationalist bias as part of the failure of AI at solving complex tasks such as understanding human languages
- In An Inquiry Concerning Human Understanding (1748), Hume’s skepticism extended to the analysis of miracles. Although Hume didn’t address the nature of miracles directly, he did question the testimony-based belief in the miraculous. This skepticism, of course, was seen as a direct threat by believers in the bible as well as many other purveyors of religious traditions. The Reverend Thomas Bayes was both a mathematician and a minister. One of his papers, called Essay towards Solving a Problem in the Doctrine of Chances (1763) addressed Hume’s questions mathematically. Bayes’ theorem demonstrates formally how, through learning the correlations of the effects of actions, we can determine the probability of their causes.

The Development of Formal Logic

- Gottfried Wilhelm von Leibniz, with his Calculus Philosophicus, introduced the first system of formal logic as well as proposed a machine for automating its tasks (Leibniz 1887). Furthermore, the steps and stages of this mechanical solution can be represented as movement through the states of a tree or graph. Leonhard Euler, in the eighteenth century, with his analysis of the “connectedness” of the bridges joining the riverbanks and islands of the city of Königsberg.
- Introduced the study of representations that can abstractly capture the structure of relationships in the world as well as the discrete steps within a computation about these relationships (Euler 1735). The formalization of graph theory also afforded the possibility of state space search, a major conceptual tool of artificial intelligence. We can use graphs to model the deeper structure of a problem.
- The nodes of a state space graph represent possible stages of a problem solution; the arcs of the graph represent inferences, moves in a game, or other steps in a

problem solution. Solving the problem is a process of searching the state space graph for a path to a solution By describing the entire space of problem solutions, state space graphs provide a powerful tool for measuring the structure and complexity of problems and analyzing the efficiency, correctness, and generality of solution strategies.

- As one of the originators of the science of operations research, as well as the designer of the first programmable mechanical computing machines, Charles Babbage, a nineteenth century mathematician, may also be considered an early practitioner of artificial intelligence (Morrison and Morrison 1961). Babbage's difference engine was a specialpurpose machine for computing the values of certain polynomial functions and was the forerunner of his analytical engine. The analytical engine, designed but not successfully constructed during his lifetime, was a general-purpose programmable computing machine.
- We may say most aptly that the Analytical Engine weaves algebraical patterns just as the Jacquard loom weaves flowers and leaves. Here, it seems to us, resides much more of originality than the difference engine can be fairly entitled to claim

The Turing Test

- One of the earliest papers to address the question of machine intelligence specifically in relation to the modern digital computer was written in 1950 by the British mathematician Alan Turing. Computing Machinery and Intelligence (Turing 1950) remains timely in both its assessment of the arguments against the possibility of creating an intelligent computing machine and its answers to those arguments. Turing, known mainly for his contributions to the theory of computability, considered the question of whether or not a machine could actually be made to think.
- The Turing test measures the performance of an allegedly intelligent machine against that of a human being, arguably the best and only standard for intelligent behavior. The test, which Turing called the imitation game, places the machine and a human counterpart in rooms apart from a second human being, referred to as the interrogator Noting that the fundamental ambiguities in the question itself (what is thinking? what is a machine?) precluded any rational answer, he proposed that the question of intelligence be replaced by a more clearly defined empirical test.

- The interrogator is not able to see or speak directly to either of them, does not know which entity is actually the machine, and may communicate with them solely by use of a textual device such as a terminal. The interrogator is asked to distinguish the computer from the human being solely on the basis of their answers to questions asked over this device. If the interrogator cannot distinguish the machine from the human, then, Turing argues, the machine may be assumed to be intelligent

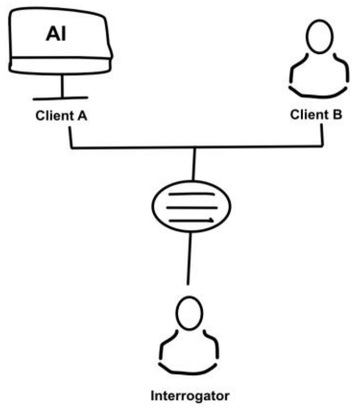

Figure 1.1 The Turing test.

The important features of Turing's test are:

1. It attempts to give an objective notion of intelligence, i.e., the behavior of a known intelligent being in response to a particular set of questions. This provides a standard for determining intelligence that avoids the inevitable debates over its "true" nature.

2. It prevents us from being sidetracked by such confusing and currently unanswerable questions as whether or not the computer uses the appropriate internal processes or whether or not the machine is actually conscious of its actions. 3. It eliminates any bias in favor of living organisms by forcing the interrogator to focus solely on the content of the answers to questions.

Because of these advantages, the Turing test provides a basis for many of the schemes actually used to evaluate modern AI programs. A program that has potentially achieved intelligence in some area of expertise may be evaluated by comparing its performance on a given set of problems to that of a human expert. This evaluation technique is just a variation of the Turing test: a group of humans are asked to blindly compare the performance of a computer and a human being on a particular set of problems. As we will see, this methodology has become an essential tool in both the development and verification of modern expert systems.

- The Turing test, in spite of its intuitive appeal, is vulnerable to a number of justifiable criticisms. One of the most important of these is aimed at its bias toward purely symbolic problem-solving tasks. It does not test abilities requiring perceptual skill or manual dexterity, even though these are important components of human intelligence. Conversely, it is sometimes suggested that the Turing test needlessly constrains machine intelligence to fit a human mold.
- Perhaps machine intelligence is simply different from human intelligence and trying to evaluate it in human terms is a fundamental mistake. Do we really wish a machine would do mathematics as slowly and inaccurately as a human? Shouldn't an intelligent machine capitalize on its own assets, such as a large, fast, reliable memory, rather than trying to emulate human cognition? In fact, a number of modern AI practitioners (e.g., Ford and Hayes 1995) see responding to the full challenge of Turing's test as a mistake and a major distraction to the more important work at hand: developing general theories to explain the mechanisms of intelligence in humans and machines and applying those theories to the development of tools to solve specific, practical problems. Although we agree with the Ford and Hayes concerns in the large, we still see Turing's test as an important component in the verification and validation of modern AI software.
- Many modern AI programs consist of a collection of modular components, or rules of behavior, that do not execute in a rigid order but rather are invoked as needed in response to the structure of a particular problem instance. Pattern matchers allow general rules to apply over a range of instances. These systems have an extreme flexibility that enables relatively small programs to exhibit a vast range of possible behaviors in response to differing problems and situations.

Biological and Social Models of Intelligence: Agents Theories

- we have approached the problem of building intelligent machines from the viewpoint of mathematics, with the implicit belief of logical reasoning as paradigmatic of intelligence itself, as well as with a commitment to "objective" foundations for logical reasoning. This way of looking at knowledge, language, and thought reflects the rationalist tradition of western philosophy, as it evolved through Plato, Galileo, Descartes, Leibniz, and many of the other philosophers discussed earlier in this chapter. It also reflects the underlying assumptions of the Turing test, particularly its emphasis on symbolic reasoning as a test of intelligence, and the belief that a straightforward comparison with human behavior was adequate to confirming machine intelligence.

- The reliance on logic as a way of representing knowledge and on logical inference as the primary mechanism for intelligent reasoning are so dominant in Western philosophy that their "truth" often seems obvious and unassailable. It is no surprise, then, that approaches based on these assumptions have dominated the science of artificial intelligence from its inception almost through to the present day.

What are the main themes supporting an agent-oriented and emergent view of intelligence? They include:

1. Agents are autonomous or semi-autonomous. That is, each agent has certain responsibilities in problem solving with little or no knowledge of either what other agents do or how they do it. Each agent does its own independent piece of the problem solving and either produces a result itself (does something) or reports results back to others in the community (communicating agent).

2. Agents are "situated." Each agent is sensitive to its own surrounding environment and (usually) has no knowledge of the full domain of all agents. Thus, an agent's knowledge is limited to the tasks to hand: "the-file-I'm-processing" or "the-wall-next-to-me" with no knowledge of the total range of files or physical constraints in the problem-solving task.

3. Agents are interactional. That is, they form a collection of individuals that cooperate on a particular task. In this sense they may be seen as a "society" and, as with human society, knowledge, skills, and responsibilities, even when seen as collective, are distributed across the population of individuals.

4. The society of agents is structured. In most views of agent-oriented problem solving, each individual, although having its own unique environment and skill set, will

coordinate with other agents in the overall problem solving. Thus, a final solution will not only be seen as collective, but also as cooperative.

5. Finally, the phenomenon of intelligence in this environment is "emergent." Although individual agents are seen as possessing sets of skills and responsibilities, the overall cooperative result can be viewed as greater than the sum of its individual contributors. Intelligence is seen as a phenomenon resident in and emerging from a society and not just a property of an individual agent.

Based on these observations, we define an agent as an element of a society that can perceive (often limited) aspects of its environment and affect that environment either directly or through cooperation with other agents. Most intelligent solutions require a variety of agents. These include rote agents, that simply capture and communicate pieces of information, coordination agents that can support the interactions between other agents, search agents that can examine multiple pieces of information and return some chosen bit of it, learning agents that can examine collections of information and form concepts or generalizations, and decision agents that can both dispatch tasks and come to conclusions in the light of limited information and processing. Going back to an older definition of intelligence, agents can be seen as the mechanisms supporting decision making in the context of limited processing resources.

The main requisites for designing and building such a society are:

1. structures for the representation of information,

2. strategies for the search through alternative solutions, and

3. the creation of architectures that can support the interaction of agents.

- Our preliminary discussion of the possibility of a theory of automated intelligence is in no way intended to overstate the progress made to date or minimize the work that lies ahead. As we emphasize throughout this book, it is important to be aware of our limitations and to be honest about our successes. For example, there have been only limited results with programs that in any interesting sense can be said to "learn". Our accomplishments in modeling the semantic complexities of a natural language such as English have also been very modest. Even fundamental issues such as organizing knowledge or fully managing the complexity and correctness of very large computer programs (such as large knowledge bases) require considerable further research. Knowledge-based systems, though they have achieved marketable engineering successes, still have many limitations in the quality and generality of their reasoning. These include their inability to perform commonsense reasoning or to exhibit knowledge of rudimentary physical reality, such as how things change over time.

AI Challenge

Knowledge representation

- Issues in representation:

The representation of information for use in intelligent problem solving offers important and difficult challenges that lie at the core of AI.

we present a brief historical retrospective of early research in representation; topics include

1.semantic networks,

2.conceptual dependencies,

3.scripts, and frames.

A brief historical retrospective of early research in representation:

Associationist Theories of Meaning:

- Logical representations grew out of the efforts of philosophers and mathematicians to characterize the principles of correct reasoning. The major concern of logic is the development of formal representation languages with sound and complete inference rules. As a result, the semantics of predicate calculus emphasizes truth-preserving operations on well-formed expressions.
- An alternative line of research has grown out of the efforts of psychologists and linguists to characterize the nature of human understanding. This work is less concerned with establishing a science of correct reasoning than with describing the way in which humans actually acquire, associate, and use knowledge of their world. This approach has proved particularly useful to the AI application areas of natural language understanding and commonsense reasoning.

- There are many problems that arise in mapping commonsense reasoning into formal logic. For example, it is common to think of the operators ∨ and → as corresponding to the English “or” and “if ... then ...”. However, these operators in logic are concerned solely with truth values and ignore the fact that the English “if ... then ...” suggests specific relationship (often more coorelational than causal) between its premises and its conclusion. For example, the sentence “If a bird is a cardinal, then it is red” (associating the bird cardinal with the color red) can be written in predicate calculus:

$$\forall X\ (\text{cardinal}(X) \rightarrow \text{red}(X)).$$

This may be changed, through a series of truth-preserving operations, logically equivalent expression

$$\forall X\ (\neg\, \text{red}(X) \rightarrow \neg\, \text{cardinal}(X)).$$

- These two expressions have the same truth value; that is, the second is true if and only if the first is true. However, truth value equivalence is inappropriate in this situation. If we were to look for physical evidence of the truth of these statements, the fact that this sheet of paper is not red and also not a cardinal is evidence for the truth of the second expression. Because the two expressions are logically equivalent, it follows that it is also evidence for the truth of the first statement. This leads to the conclusion that the whiteness of the sheet of paper is evidence that cardinals are red.
- This line of reasoning strikes us as meaningless and rather silly. The reason for this incongruity is that logical implication only expresses a relationship between the truth values of its operands, while the English sentence implied a positive coorelation between membership in a class and the possession of properties of that class. In fact, the genetic makeup of a bird causes it to have a certain color. This relationship is lost in the second version of the expression. Although the fact that the paper is not red is consistent with the truth of both sentences, it is irrelevant to the causal nature of the color of birds.

Semantic Net:

- Network representations have almost as long a history as logic. The Greek philosopher Porphyry created tree-based type hierarchies - with their roots at the top - to describe Aristotle’s categories (Porphyry 1887). Frege developed a tree notation for logic expressions. Perhaps the earliest work to have a direct influence on contemporary semantic nets was Charles S. Peirce's system of existential graphs, developed in the nineteenth century (Roberts 1973). Peirce's theory had all the expressive power of first-order predicate calculus, with an axiomatic basis and formal rules of inference.

- Graphs have long been used in psychology to represent structures of concepts and associations. Selz (1913, 1922) pioneered this work, using graphs to represent concept hierarchies and the inheritance of properties. He also developed a theory of schematic anticipation that influenced AI work in frames and schemata. Anderson, Norman, Rumelhart, and others have used networks to model human memory and intellectual performance (Anderson and Bower 1973, Norman et al. 1975).
- Much of the research in network representations has been done in the arena of natural language understanding. In the general case, language understanding requires an understanding of common sense, the ways in which physical objects behave, the interactions that occur between humans, and the ways in which human institutions are organized. A natural language program must understand intentions, beliefs, hypothetical reasoning, plans, and goals. Because of these requirements language understanding has always been a driving force for research in knowledge representation.

The numbers in the response indicate that the program has selected from among different meanings of the words.

This approach to semantics might provide a natural language understanding system with the ability to:

1. Determine the meaning of a body of English text by building up collections of these intersection nodes.

2. Choose between multiple meanings of words by finding the meanings with the shortest intersection path to other words in the sentence. For example, it could select a meaning for "plant" in "Tom went home to water his new plant" based on the intersection of the word concepts "water" and "plant."

3. Answer a flexible range of queries based on associations between word concepts in the queries and concepts in the system.

Although this and other early work established the power of graphs to model associative meaning, it was limited by the extreme generality of the formalism. Knowledge was generally structured in terms of specific relationships such as object/property, class/subclass, and agent/verb/object.

Scripts:

A natural language understanding program must use a large amount of supporting knowledge to understand even the simplest conversation There is evidence that humans organize this knowledge into structures corresponding to typical situations (Bartlett 1932). If we are reading a story about restaurants, baseball, or politics, we resolve any ambiguities in the text in a way consistent with restaurants, baseball, or politics. If the subject of a story changes abruptly, there is evidence that people pause briefly in their reading, presumably to change knowledge structures. It is hard to understand a poorly organized or structured story, possibly because we cannot easily fit it into any of our existing knowledge structures. There can also be errors in understanding when the subject of a conversation changes abruptly, presumably because we are confused over which context to use in resolving pronoun references and other ambiguities in the conversation.

A script is a structured representation describing a stereotyped sequence of events in a particular context. The script was originally designed by Schank and his research group as a means of organizing conceptual dependency structures into descriptions of typical situations. Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system is to understand.

The components of a script are:

- Entry conditions or descriptors of the world that must be true for the script to be called. In our example script, these include an open restaurant and a hungry customer that has some money.
- Results or facts that are true once the script has terminated; for example, the customer is full and poorer, the restaurant owner has more money
- Props or the “things” that support the content of the script. These will include tables, waiters, and menus. The set of props supports reasonable default assumptions about the situation: a restaurant is assumed to have tables and chairs unless stated otherwise

- Roles are the actions that the individual participants perform. The waiter takes orders, delivers food, and presents the bill. The customer orders, eats, and pays.
- Scenes. Schank breaks the script into a sequence of scenes each of which presents a temporal aspect of the script. In the restaurant there is entering, ordering, eating, etc.

The elements of the script, the basic “pieces” of semantic meaning, are represented using conceptual dependency relationships. Placed together in a framelike structure, they represent a sequence of meanings, or an event sequence.

Frames:

- Another representational scheme, in many ways similar to scripts, that was intended to capture in explicitly organized data structures the implicit connections of information in a problem domain, was called frames.
- This representation supports the organization of knowledge into more complex units that reflect the organization of objects in the domain. In a 1975 paper, Minsky describes a frame

Each individual frame may be seen as a data structure, similar in many respects to the traditional “record”, that contains information relevant to stereotyped entities. The slots in the frame contain information such as

1. Frame identification information
2. Relationship of this frame to other frames
3. Descriptors of requirements for a frame
4. Procedural information on use of the structure described.
5. Frame default information
6. New instance information

- Frames extend semantic networks in a number of important ways. Although the frame description of hotel beds, , the frame version makes it much clearer that we are describing a bed with its various attributes. In the network version, there is simply a collection of nodes and we depend more on our interpretation of the structure to see the hotel bed as the primary object being described. This ability to organize our knowledge into such structures is an important attribute of a knowledge base.
- Procedural attachment is an important feature of frames because it supports the linking of specific pieces of code to appropriate entities in the frame representation. For

example, we might want to include the ability to generate graphic images in a knowledge base. A graphics language is more appropriate for this than a network language. We use procedural attachment to create demons. A demon is a procedure that is invoked as a side effect of some other action in the knowledge base. For example, we may wish the system to perform type checks or to run consistency tests whenever a certain slot value is changed.

Conceptual graphs

The early research work in AI that developed representational schemes.) a number of network languages were created to model the semantics of natural language and other domains.

✓ Introduction to Conceptual Graphs

- A conceptual graph is a finite, connected, bipartite graph. The nodes of the graph are either concepts or conceptual relations. Conceptual graphs do not use labeled arcs; instead, the conceptual relation nodes represent relations between concepts. Because conceptual graphs are bipartite, concepts only have arcs to relations, and vice versa. In Figure 7.14 dog and brown are concept nodes and color a conceptual relation.
- In conceptual graphs, concept nodes represent either concrete or abstract objects in the world of discourse. Concrete concepts, such as a cat, telephone, or restaurant, are characterized by our ability to form an image of them in our minds. Note that concrete concepts include generic concepts such as cat or restaurant along with concepts of specific cats and restaurants. We can still form an image of a generic cat. Abstract concepts include things such as love, beauty, and loyalty that do not correspond to images in our minds.
- Conceptual relation nodes indicate a relation involving one or more concepts. One advantage of formulating conceptual graphs as bipartite graphs rather than using labeled arcs is that it simplifies the representation of relations of any arity.

Types, Individuals, and Names

- Many early designers of semantic networks were careless in defining class/member and class/subclass relationships, with resulting semantic confusion. For example, the relation between an individual and its class is different from the relation between a class (such as dog) and its superclass (carnivore). Similarly, certain properties belong to individuals, and others belong to the class itself; the representation should provide a vehicle for making this distinction. The properties of having fur and liking bones belong to individual dogs; the class "dog" does not have fur or eat anything. Properties that are appropriate to the class include its name and membership in a zoological taxonomy.

- In conceptual graphs, every concept is a unique individual of a particular type. Each concept box is labeled with a type label, which indicates the class or type of individual represented by that node. Thus, a node labeled dog represents some individual of that type. Types are organized into a hierarchy. The type dog is a subtype of carnivore, which is a subtype of mammal, etc. Boxes with the same type label represent concepts of the same type; however, these boxes may or may not represent the same individual concept.
- Each concept box is labeled with the names of the type and the individual. The type and individual labels are separated by a colon, ":". The graph of Figure 7.16 indicates that the dog "Emma" is brown. The graph of Figure 7.17 asserts that some unspecified entity of type dog has a color of brown. If the individual is not indicated, the concept represents an unspecified individual of that type.

The Type Hierarchy

- The type hierarchy, as illustrated by Figure 7.21, is a partial ordering on the set of types, indicated by the symbol. If s and t are types and t s, then t is said to be a subtype of s and s is said to be a supertype of t. Because it is a partial ordering, a type may have one or more supertypes as well as one or more subtypes. If s, t, and u are types, with t s and t u, then t is said to be a common subtype of s and u. Similarly, if s v and u v then v is a common supertype of s and u.
- The type hierarchy of conceptual graphs forms a lattice, a common form of multiple inheritance system. In a lattice, types may have multiple parents and children. However, every pair of types must have a minimal common supertype and a maximal common subtype. For types s and u, v is a minimal common supertype if s v , u v, and for any w, a common supertype of s and u, v w. Maximal common subtype has a corresponding definition. The minimal common supertype of a collection of types is the appropriate place to define properties common only to those types. Because many types, such as emotion and rock, have no obvious common supertypes or subtypes, it is necessary to add types that fill these roles. To make the type hierarchy a true lattice, conceptual

graphs include two special types. The universal type, indicated by T, is a supertype of all types. The absurd type, indicated by ⊥, is a subtype of all types.

Generalization and Specialization

- The theory of conceptual graphs includes a number of operations that create new graphs from existing graphs. These allow for the generation of a new graph by either specializing or generalizing an existing graph, operations that are important for representing the semantics of natural language. The four operations are copy.

Restrict allows concept nodes in a graph to be replaced by a node representing their specialization. There are two cases:

1. If a concept is labeled with a generic marker, the generic marker may be replaced by an individual marker.
2. 2. A type label may be replaced by one of its subtypes, if this is consistent with the referent of the concept. we can replace animal with dog.
   The join rule lets us combine two graphs into a single graph. If there is a concept node c1 in the graph s1 that is identical to a concept node c2 in s2, then we can form a new graph by deleting c2 and linking all of the relations incident on c2 to c1. Join is a specialization rule, because the resulting graph is less general than either of its components.
3. If a graph contains two duplicate relations, then one of them may be deleted, along with all its arcs. This is the simplify rule. Duplicate relations often occur as the result of a join operation.

Propositional Nodes:

- In addition to using graphs to define relations between objects in the world, we may also want to define relations between propositions. Consider, for example, the statement “Tom believes that Jane likes pizza”. “Believes” is a relation that takes a proposition as its argument.
- Conceptual graphs include a concept type, proposition, that takes a set of conceptual graphs as its referent and allows us to define relations involving propositions. Propositional concepts are indicated as a box that contains another conceptual graph. These proposition concepts may be used with appropriate relations to represent knowledge about propositions

- how conceptual graphs with propositional nodes may be used to express the modal concepts of knowledge and belief. Modal logics are concerned with the various ways propositions are entertained: believed, asserted as possible, probably or necessarily true, intended as a result of an action, or counterfactual.
- Using conceptual graphs, we can easily represent conjunctive concepts such as "The dog is big and hungry", but we have not established any way of representing negation or disjunction. Nor have we addressed the issue of variable quantification.

Alternative representations and ontologies.

- AI researchers have continued to question the role of explicit representation in intelligence. Besides the connectionist and emergent approaches of Chapters 11 and 12. a further challenge to the role of traditional representation comes from Rodney Brooks' work at MIT.
- The need for any centralized representational scheme, and with his subsumption architecture, attempts to show how general intelligence might evolve from lower and supporting forms of intelligence.
- A second approach to the problem of explicit and static representations comes from the work of Melanie Mitchell and Douglas Hofstadter at Indiana University. The Copycat architecture is an evolving network which adjusts itself to the meaning relationships that it encounters through experiment with an external world.
- Finally, when building problem solvers in complex environments it is often necessary to employ multiple different representational schemes. Each representational scheme may be referred to as an ontology. It then becomes necessary to build communication and other links between these different representations to support knowledge management, communication, backup, and other aspects of the problem solving. We introduce these issues and possible solutions under the general heading of knowledge management technology.

Brooks' Subsumption Architecture

Brooks conjectures, and offers examples through his robotic creations, that intelligent behavior does not come from disembodied systems like theorem provers, or even from traditional expert systems Intelligence, Brooks claims, is the product of the interaction between an appropriately designed system and its environment. Furthermore, Brooks espouses the view that intelligent behavior emerges from the interactions of architectures of organized simpler behaviors: his subsumption architecture

The subsumption architecture is a layered collection of task-handlers. Each task is accomplished by a finite state machine that continually maps a perception-based input into an action-oriented output. This mapping is accomplished through simple sets of condition → action production rules These rules determine, in a fairly blind fashion, that is, with no global state knowledge, what actions are appropriate to the current state of that subsystem. Brooks does allow some feedback to lower level systems.

, A number of important questions concerning the subsumption architecture and related approaches to the design of control systems.

1. There is a problem of the sufficiency of information at each system level. Since at each level, purely reactive state machines make decisions on local information, it is difficult to see how such decisions can take account of information not at that level. By definition, it will be myopic.

2. If there exists absolutely no "knowledge" or "model" of the complete environment, how can the limited input on the local situation be sufficient for determination of globally acceptable actions? How can top level coherence result?

3. How can a purely reactive component with very limited state learn from its environment? At some level in the system, there must be sufficient state for the creation of learning mechanisms if the overall agent is to be called intelligent.

4. There is a problem of scale. Although Brooks and his associates claim to have built subsumption architectures of six and even ten layers, what design principles allow it to scale to further interesting behavior, i.e., to large complex systems?

Copycat

- An often-heard criticism of traditional AI representation schemes is that they are static and cannot possibly reflect the dynamic nature of thought processes and intelligence. When a human perceives a new situation, for example, he or she is often struck by relationships with already known or analogous situations.

- Copycat is made up of three components, the workspace, the slipnet, and the coderack. These three are mediated in their interactions by a temperature measure. The temperature captures the degree of perceptual organization in the system, and controls the degree of randomness used in making decisions. Higher temperatures reflect the fact that there is little information on which to base decisions and thus they are more random. A temperature drop indicates the system is building consensus and a low temperature indicates that an answer is emerging and reflects the program's "confidence" in that solution.
- The workspace is a global structure, similar to the blackboard of Section 6.3, for creating structures that the other components of the system can inspect. In this sense it is also much like the message area in Holland's (1986) classifier system. The workspace is where perceptual structures are built hierarchically on top of the input and gives possible states for the workspace, with bonds (the arrows) built between related components of the strings.

The slipnet reflects the network of concepts or potential associations for the components of the analogy. One view of the slipnet is as a dynamically deformable semantic network, each of whose nodes has an activation level. Links in the network can be labeled by other nodes. Based on the activation level of the labelling nodes, the linked nodes grow or shrink. In this way the system changes the degree of association between the nodes as a function of context. The coderack is a priority based probabilistic queue containing codelets. Codelets are small pieces of executable code designed to interact with the objects in the workspace and to attempt to further some small part of the evolving solution, or, more simply, to explore different facets of the problem space.

Multiple Representations, Ontologies, and Knowledge Services

- Many complex AI applications require that multiple representational schemes be designed, each set up for specific casks, and then integrated into a user-oriented software platform. A simple example of this might be the creation of a virtual meeting space. The various attendees, from their remote locations would interact with each other in this virtual "space". Besides voice technology, the virtual meeting space would require video streaming, text and presentation sharing, records access, and so on. One approach to this task would be to adapt various off-the-shelf components and then integrate them into a knowledge service that supports the interactions required.

- The automated virtual meeting service just described is an example of integrating multiple representational schemes to provide a service. We often use the term ontology to characterize the representation scheme of a particular component of that service. The resulting knowledge service requires the integration of these components. Ontology is derived from a Greek word, a form of the verb "to be", that means being or existence. Thus, the term refers to the components of a software module and the structure supporting the interactions of these components, that is, to their function or "constituent being" as part of the module. Agent-Oriented Problem Solving: A Definition

<u>Agent-Oriented Problem Solving: A Definition</u>

- An autonomous system is one that can interact with its environment without the direct intervention of other agents. To do this it must have control over its own actions and internal state. Some autonomous agents can also learn from their experience to improve their performance over time (see machine learning, Part IV). For example, on the internet, an autonomous agent could do a credit card authentication check independent of other issues in the purchasing transaction. In the ROBOCUP example, an agent could pass the ball to a teammate or kick it on goal depending on its individual situation
- A flexible agent is both intelligently responsive as well as proactive depending on its current situation. A responsive agent receives stimuli from its environment and responds to them in an appropriate and timely fashion. A proactive agent does not simply respond to situations in its environment but is also able to plan, be opportunistic, goal directed, and have appropriate alternatives for various situations. A credit agent, for example, would be able to go back to the user with ambiguous results or find another credit agency if one alternative is not sufficient. The soccer agent could change its trajectory depending on the challenge pattern of an opponent.

An agent is social that can interact, as appropriate, with other software or human agents. After all, an agent is only part of a complex problem-solving process. The interactions of the social agent are oriented towards the goals of the larger multi-agent system. This social dimension of the agent system must address many difficult situations. How can different agents bid for a subtask in problem solving? How can agents communicate with each other to facilitate the accomplishment of higher systemlevel tasks - in the ROBOCUP example, this might be to score a goal? How can one agent support another agent's goals, for example, to handle the security issues of an internet task? All these questions on the social dimension are the subject of ongoing research.

Examples of and Challenges to an Agent-Oriented Paradigm

- To make the ideas of the previous section more concrete we next describe a number of application domains where agent-based problem solving is appropriate. We also include references to research within these problem areas.
- Manufacturing The manufacturing domain can be modeled as a hierarchy of work areas. There may be work areas for milling, lathing, painting, assembling, and so on. These work areas may then be grouped into manufacturing subsystems, each subsystem a function within the larger manufacturing process.

- Automated Control Since process controllers are autonomous, reactive, and often distributed systems, it is not surprising that agent models can be important.

Telecommunications Telecommunication systems are large distributed networks of interacting components that require real-time monitoring and management. Agentbased systems have been used for network control and management (Schoonderwoerd et al. 1997, Adler et al. 1989, Fatima et al. 2006), transmission and switching (Nishibe et al. 1993), and service (Busuoic and Griffiths 1994). See (Veloso et al. 2000) for a comprehensive overview.

- Transportation Systems Traffic systems are almost by definition distributed, situated, and autonomous. Applications include coordinating commuters and cars for carpooling (Burmeister et al. 1997) and cooperative transportation scheduling (Fischer et al. 1996).
- Information Management The richness, diversity, and complexity of information available to current society is almost overwhelming. Agent systems allow the possibility of intelligent information management, especially on the internet. Both human factors as well as information organization seem to conspire against comfortable access to information.