AI Assignment 09

Propositional logic:

Knowledge based agents: Agents whose intellligence is based on reasoning that operates on internal representations of knowledge.

a. an implication is true if both premise and conclusion are true, or if its premise is false.

- Knowlwdge Base(KB) is the central component of a knowledge based agent.
- Knowledge Base is a set of sentences, written in a knowledge representation language.(axioms are sentences that are not derived from any other sentence)
- Inference: deriving new sentences from old.
- TELL and ASK methods to update a knowledge base with new sentences and ask what it already knows.
- knowlege level, implementation level.
- Declarative approach Starting with an empty knowledge base, the agent designer can TELL sentences one by one (DECLARATIVE) until the agent knows how to operate in its environment.
- procedural approach encodes desired behaviors directly as program code.
- syntax specifies a well formed sentence. Sentences must be specified according to the syntax of the represntation language.
- Semantics(meaning of the sentences) defines truth of each sentence with respect to each possible-world(models, Models are assignments of true or false to every proposition symbol). every sentence must be either true or false.
- Logical Entailment(or Logical consequence) - the idea that a sentence follows logically from another sentence.
- An inference algorithm that derives only entailed sentences is called sound or TRUTH-PRESERVING.
- an inference algorithm is complete if it can derive any sentence that is entailed.(Completeness)
- The atomic sentences: consist of a single proposition symbol. each symbol stands for a propositon that can be true or false.

- Complex sentences are constructed from simpler sentences, using parentheses and logical connectives.
- NEGATION ¬ (not)
- ∧ (and). CONJUNCTION
- DISJUNCTION ∨ (or)
- IMPLICATION ⇒ (Implications are also known as rules or if–then statements.)
- BICONDITIONAL ⇔
- OPERATOR PRECEDENCE : ¬, ∧, ∨, ⇒, ⇔
- In propositional logic, a model simply fixes the truth value—true or false—for every proposition symbol.
- theorem proving—applying rules of inference directly to the sentences in our knowledge base to construct a proof of the desired sentence without consulting models.
- validity: A sentence is valid if it is true in all models.Valid sentences are also known as tautologies—they are necessarily true.
- satisfiability: A sentence is satisfiable if it is true in,(or satisfied by,) some model.
- proof by refutation or proof by contradiction. One assumes a CONTRADICTION sentence β to be false and shows that this leads to a contradiction with known axioms α.
- inference rules that can be applied to derive a proof—a chain of conclusions that leads to the desired goal. (eg :Modus Ponens, And-Elimination)
- monotonicity: says that the set of entailed sentences can only increase as information is added to the knowledge base.(another property of logical systems)
- resolutio: a single inference rule, that yields a complete inference algorithm when coupled with any complete search algorithm.(resolution, resolvent,unit resolution inference rule)
- complementary literals (i.e., one is the negation of the other).
- the unit resolution rule takes a clause—a disjunction of literals—and a literal and produces a new clause. Note that a single literal can be viewed as a

disjunction of one literal, also known as a unit clause.

- There is one more technical aspect of the resolution rule: the resulting clause should contain
- only one copy of each literal. The removal of multiple copies of literals is called factoring
- sentence expressed as a conjunction of clauses is said to be in conjunctive normal form or CNF
- The completeness theorem for resolution in propositional logic is called the ground
- resolution theorem: If a set of clauses is unsatisfiable, then the resolution closure of those clauses contains the empty clause.
- definite clause, which is a disjunction of literals of which exactly one is positive.
- Horn clause, which is a disjunction of literals of which atmost one is positive.
- clauses with no positive literals; these are called goal clauses.
- In Horn form, the premise is called the body and the conclusion is called the head. sentence consisting of a single positive literal, such as L1,1, is called a fact.
- data-driven reasoning—that is, reasoning in which the focus of attention starts with the known data.
- Backward chaining and forward chaining.
- forward chaining is a form of data-directed reasoning.
- Backward chaining is a form of goal-directed reasoning

First Order Logic:

- Models for first-order logic have objects in them! The domain of a model is the set of objects or domain elements it contains.
- The domain is required to be nonempty—every possible world must contain at least one object.
- a relation is just the set of tuples of objects that are related.(A tuple is a collection of objects arranged in a fixed order and is written with angle brackets

surrounding the objects.)

- relations are binary relations—that is, they relate pairs of objects. The model also contains unary relations, or properties.
- models in first-order logic require total functions, that is, there must be a value for every input tuple.
- symbols, come in three kinds: constant symbols: which stand for objects; predicate symbols: which stand for relations; and function symbols: which stand for functions.
- Each predicate and function symbol comes with an arity that fixes the number of arguments.
- each model includes an interpretation that specifies exactly which objects, relations and functions are referred to by the constant, predicate, and function symbols.
- operator precedence: ¬, =, ∧, ∨, ⇒, ⇔
- A term is a logical expression that refers to an object. Constant symbols are therefore terms.
- a complex term is formed by a function symbol followed by a parenthesized list of terms as arguments to the function symbol. a complex term is just a complicated kind of name. It is not a "subroutine call" that "returns a value."the term as a whole refers to the object that is the value of the function F applied to d1,...,dn.
- sentence (or atom for short) is formed from a predicate symbol optionally followed by a parenthesized list of terms, such as :Brother (Richard, John)
- An atomic sentence is true in a given model if the relation referred to by the predicate symbol holds among the objects referred to by the arguments.
- We can use logical connectives to construct more complex sentence, eg: ¬Brother (LeftLeg(Richard), John), Brother (Richard, John) ∧ Brother (John, Richard)
- Quantifiers let us express properties of entire collections of objects.First-order

logic contains two standard quantifiers, called universal and existential.

- Universal quantifier: $\forall$ x King(x) $\Rightarrow$ Person(x) . $\forall$ is usually pronounced "For all ...". Universal quantification makes statements about every object. (A variable is a term all by itself,and as such can also serve as the argument of a function—for example, LeftLeg(x). A term with no variables is called a ground term.)

- Extended interperetations: extended interpretation specifies a domain element to which x refers.

- Existential Quantifier: $\exists$ x Crown(x) $\wedge$ OnHead(x, John) . $\exists$x is pronounced "There exists an x such that ..." or "For some x...". Intuitively, the sentence $\exists$ x P says that P is true for at least one object x. More precisely, $\exists$ x P is true in a given model if P is true in at least one extended interpretation that assigns x to a domain element.

- Just as $\Rightarrow$ appears to be the natural connective to use with $\forall$, $\wedge$ is the natural connectiveto use with $\exists$.

- Nested quantifiers: eg: $\forall$ x $\exists$ y Loves(x, y) ; $\forall$ x, y Sibling(x, y) $\Leftrightarrow$ Sibling(y, x) .

- Connections between $\forall$ and $\exists$: $\forall$ x $\neg$Likes(x,Parsnips) is equivalent to $\neg\exists$ x Likes(x,Parsnips).

- First-order logic includes one more way to make atomic sentences, other than using a predicate and terms as described earlier. We can use the equality symbol to signify that two terms refer to the same object. For example:Father (John) = Henry.

- Database semantics: every constant symbol refer to a distinct object—the so-called unique-names assumption; atomic sentences not known to be true are in fact false—the closed-world assumption; domain closure, meaning that each model contains no more domain elements than those named by the constant symbols.

- there is no one "correct" semantics for logic. The usefulness of any proposed semantics depends on how concise and intuitive it makes the expression of the kinds of knowledge we want to write down, and on how easy and natural it is to develop the corresponding rules of inference.

- Database semantics is most useful when we are certain about the identity of all the objects described in the knowledge base and when we have all the facts at hand

- In knowledge representation, a domain is just some part of the world about which we wish to express some knowledge.
- Sentences are added to a knowledge base using TELL, Such sentences are called assertions
- Questions asked with ASK are called queries or goals. Generally speaking, any query that is logically entailed by the knowledge base should be answered affirmatively.
- If we want to know what value of x makes the sentence true, we will need a different function, ASKVARS, which we call with:ASKVARS(KB,Person(x)) and which yields a stream of answers, Such an answer is called a substitution or binding list.
- Axioms: They provide the basic factual information from which useful conclusions can be derived.
- Numbers, sets, and lists The Peano axioms define natural numbers and addition.
- syntactic sugar, that is, an extension to or abbreviation of the standard syntax that does not change the semantics.(Any sentence that uses sugar can be “desugared” to produce an equivalent sentence in ordinary first-order logic)
- Lists are similar to sets. The differences are that lists are ordered and the same element can appear more than once in a list
- the general process of knowledge-base construction—a process called knowledge engineering
- The knowledge-engineering process:Identify the task; Assemble the relevant knowledge; Decide on a vocabulary of predicates, functions, and constants; Encode general knowledge about the domain; Encode a description of the specific problem instance; Pose queries to the inference procedure and get answers; Debug the knowledge base
- knowledge acquisition: collecting knowledge from a group of experts, by the knowledge engineer.

- ontology means a particular theory of the nature of being or existence. The ontology determines what kinds of things exist, but does not determine their specific properties and interrelationships.