FIRST ORDER LOGIC AND REASONING

HABIB ULLAH QAMAR
MSCS(SE)

FIRST ORDER PREDICATE LOGIC

- Propositional logic lacks the expressive power to *concisely* describe an environment with many objects.
- We can adopt the **foundation of propositional logic**—a declarative, compositional semantics that is context-independent and unambiguous—and build a more expressive logic on that foundation, borrowing **representational ideas from natural language** while avoiding its drawbacks.
- Noun phrases that refer to **objects**
- Verb phrases that refer to **relations** among objects
- Relations are **functions—relations** in which there is only one “value” for a given “input.”

FIRST ORDER PREDICATE LOGIC

- **Objects:** people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . . PROPERTY
- **Relations:** these can be unary relations or P**roperties** such as red, round, bogus, prime, multistoried . . ., or more general n-ary relations such as brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . .
- **Functions**: father of, best friend, third inning of, one more than, beginning of

EXAMPLES

- "One plus two equals three."
- **Objects: one, two, three, one plus two**; Relation: **equals**; Function: plus. ("One plus two" is a name for the object that is obtained by applying the function "**plus**" to the objects "one" and "two." "Three" is another name for this object.)
- Evil King John ruled England in 1200."
- **Objects**: John, England, 1200; **Relation**: ruled; **Properties**: evil, king

FIRST ORDER PREDICATE LOGIC

- First order predicate logic is the simplest form of predicate logic.
- It is more expressive to represent a good deal of our common sense knowledge than propositional logic.
- It forms the foundation of many other representation languages and
  has been studied intensively for many decades.
- Also called first-order predicate calculus, sometimes abbreviated as FOL or FOPC

FORMAL LANGUAGE

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown

SYNTAX AND SEMANTICS OF FOL

- First order predicate logic is the simplest form of predicate logic. The main types of symbols used are
- Constants are used to name specific objects or properties, e.g. Ali, Ayesha, blue, ball.
- Predicates: A fact or proposition is divided into two parts
  - Predicate: the assertion of the proposition
  - Argument: the object of the proposition
- For example, the proposition “Ali likes bananas” can be represented in predicate logic as Likes (Ali, bananas), where Likes is the predicate and Ali and bananas are the arguments.

SYNTAX AND SEMANTICS OF FOL

- **Variables:** Variables are used to a represent general class of objects/properties, e.g. in the predicate likes (X,Y), X and Y are variables that assume the values X=Ali and Y=bananas
- Formulae: Formulas combine predicates and quantifiers to represent information

SYNTAX AND SEMANTICS OF FOL

man(ahmed) father(ahmed, belal) brother(ahmed, chand) owns(belal, car) tall(belal) hates(ahmed, chand) family()	**Predicates**
∀ Y (¬sister(Y,ahmed)) ∀X,Y,Z(man(X) ∧ man(Y) man(Z) ∧ father(Z,Y) ∧ father(Z,X) ⇒ brother(X,Y))	**Formulae**
X, Y and Z	**Variables**
ahmed, belal, chand and car	**Constants**

SYNTAX AND SEMANTICS OF FOL

- The predicate section outlines the known facts about the situation in the form of predicates, i.e. predicate name and its arguments.
- So, man(ahmed) means that ahmed is a man, hates(ahmed, chand) means that ahmed hates chand.
- The formulae sections outlines formulae that use **universal quantifiers** and variables to define certain rules. $\forall Y\ (\neg sister(Y, ahmed))$ For all Y Ahmed has no sister i.e. ahmed has no sister.

SYNTAX AND SEMANTICS OF FOL (FIG 8,3 FOR MORE DETAILS

- Similarly, **∀X, Y,Z(man(X) ∧ man(Y) man(Z) ∧ father(Z,Y) ∧ father(Z,X) ⇒ brother(X,Y))**
- It means that if there are three men, X, Y and Z, and Z is the father of both X and Y, then X and Y are bothers.
- This expresses the rule for the two individuals being brothers.

KNOWLEDGE ENGINEERING IN FOL

- Knowledge-base construction— KNOWLEDGE ENGINEERING a process called **knowledge engineering**. A knowledge engineer is someone who investigates a particular domain, learns what concepts are important in that domain, and creates a formal representation of the objects and relations in the domain.
- ***A special-purpose*** knowledge bases whose domain is carefully constrained and whose range of queries is known in advance.
- ***General-purpose*** knowledge bases, which cover a broad range of human knowledge and are intended to support tasks such as natural language understanding.

THE KNOWLEDGE-ENGINEERING PROCESS

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on a vocabulary of predicates, functions, and constants.
4. Encode general knowledge about the domain
5. Encode a description of the specific problem instance
6. Pose queries to the inference procedure and get answers.
7. Debug the knowledge base.

IDENTIFY THE TASK

- KE must outline the range of questions that the KB will support and the kinds of facts that will be available for each specific problem instance.
- For example, does the circuit in Figure 8.6 actually add properly?
- If all the inputs are high, what is the output of gate A2?
- Questions about the circuit's structure are also interesting.
- For example, what are all the gates connected to the first input terminal?
- Does the circuit contain feedback loops?

ASSEMBLE THE RELEVANT KNOWLEDGE REPRESENTATION - FACTS

- The knowledge engineer might already be an expert in the domain, or might need to work with real experts to extract what they know process called **knowledge acquisition**.
- At this stage, the knowledge is not represented formally. The idea is to understand **the scope of the knowledge base,** as determined by the task, and to **understand how the domain actually works.**

DECIDE ON A VOCABULARY

- That is, translate the important domain-level concepts into logic-level names.
- This involves many questions of knowledge-engineering *style*? Once the choices have been made, the result is a vocabulary that is known as the **ontology** of the domain.
- The word *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**.

ENCODE GENERAL KNOWLEDGE ABOUT THE DOMAIN

- The knowledge engineer writes down the **axioms(functions)** for all the vocabulary terms.
- This pins down (to the extent possible) the meaning of the terms, enabling the expert to check the content.
- Often, this step reveals misconceptions or gaps in the vocabulary that must be fixed by returning to step 3 and iterating through the process.

2. The signal at every terminal is either 1 or 0:

$$\forall t \;\; Terminal(t) \;\Rightarrow\; Signal(t) = 1 \lor Signal(t) = 0 \;.$$

3. Connected is commutative:

$$\forall t_1, t_2 \;\; Connected(t_1, t_2) \;\Leftrightarrow\; Connected(t_2, t_1) \;.$$

ENCODE A DESCRIPTION OF THE SPECIFIC PROBLEM INSTANCE.

- If the ontology is well thought out, this step will be easy. It will involve writing simple atomic sentences about instances of concepts that are already part of the ontology.

$$
\begin{aligned}
&Circuit(C_1) \land Arity(C_1, 3, 2)\\
&Gate(X_1) \land Type(X_1) = XOR\\
&Gate(X_2) \land Type(X_2) = XOR\\
&Gate(A_1) \land Type(A_1) = AND\\
&Gate(A_2) \land Type(A_2) = AND\\
&Gate(O_1) \land Type(O_1) = OR\ .
\end{aligned}
$$

POSE QUERIES TO THE INFERENCE PROCEDURE AND GET ANSWERS

- This is where the reward is: we can let the **inference procedure operate** on the **axioms** and **problem-specific facts** to **derive the facts** we are interested in knowing.
- Thus, we avoid the need for writing an application-specific solution algorithm.

DEBUG THE KNOWLEDGE BASE

- Alas, the answers to queries will seldom be correct on the first try.
- More precisely, the answers will be correct *for the knowledge base as written*, assuming that the inference procedure is sound, but they will not be the ones that the user is expecting.
- For example, if an axiom is missing, some queries will not be answerable from the knowledge base.
- A considerable debugging process could ensue. Missing axioms or axioms that are too weak can be easily identified by noticing places where the chain of reasoning stops unexpectedly.

STUDY MATERIAL

- Chapter 7 + 8
- Slides at TheITeducation.com

THANKS ……

HABIB ULLAH QAMAR