Knowledge Representation using Predicate Logic

Lecture 4

IT/T/414A

Winter Semester 2014

Instructor: Kartick Ch. MONDAL

Aim of this Lecture

- To show how logic can be used to form representations of the world
- How a process of inference can be used to derive new representations about the world
- How these can be used by an intelligent agent to deduce what to do.

Aim of this Lecture

- We require:
  - A *formal language* to represent knowledge in a computer tractable form.
  - *Reasoning* - Processes to manipulate this knowledge to deduce non-obvious facts.

Introduction: Logic

- Logic is a form of knowledge representation
- Sentences have to obey syntactic laws
- Truth is established with respect to a *model* of the world and an *interpretation* which maps symbols to world objects
- Propositional logic is a very simple form of logic
- The truth table method is a sound and complete inference method, which checks truth in all possible models.
- But the truth table method is very inefficient! 2^n models for n literals
- Today: inference methods that use *syntactic operations* on sentences

Introduction: Logic

- The main challenge is to design a language which allows one to represent all the necessary knowledge
- We need to be able to make statements about the world such as describing things - people, houses, theories etc; relations between things and properties of things.

Introduction: Logic

- Logic is form of knowledge representation.
- Representing knowledge using logic is appealing because you can derive new knowledge from old mathematical deduction.
- In this formalism you can conclude that a new statement is true if by proving that it follows from the statement that are already known.
- It provides a way of deducing new statements from old ones.

Introduction: Logic

- Compared to
  - Natural languages
    - expressive but context sensitive
  - Programming languages
    - good for concrete data structures but not expressive
- Logic combines the advantages of natural languages and formal languages.
- A Logic is:
  - concise
  - unambiguous
  - context insensitive
  - expressive
  - effective for inferences

Introduction: Logic

- Many ways to translate between languages
  - A statement can be represented in different logics
  - And perhaps differently in same logic
- **Expressiveness** of a logic
  - How much can we say in this language?
- Not to be confused with logical reasoning
  - Logics are languages, reasoning is a process (may **use** logic)

Defining Logic

- A logic is defined by:
  - **Syntax**
    - Describes the possible configurations that constitute sentences.
    - Rules for constructing legal sentences in the logic
    - Which symbols we can use (English: letters, punctuation)
    - How we are allowed to combine symbols
  - **Semantics**
    - Determines what facts in the world the sentences refer to i.e. the interpretation. Each sentence makes a claim about the world.
    - How we interpret (read) sentences in the logic
    - Assigns a meaning to each sentence

Defining Logic

  - **Proof theory** - set of rules for generating new sentences that are necessarily true given that the old sentences are true.
- The relationship between sentences is called **entailment**.
- The semantics link these sentences (representation) to facts of the world.
- The proof can be used to determine new facts which follow from the old.

Defining Logic

- Example: “All students stay in hostels”
  - A valid sentence (syntax)
  - And we can understand the meaning (semantics)
  - This sentence happens to be false (there is a counter example)
- We will consider two kinds of logic:
  - **Propositional logic**
  - **First-order logic** or first-order **predicate calculus**.
- Propositional logic is of limited expressiveness but is useful to introduce many of the concepts of logic's syntax, semantics and inference procedures.

Propositional Logic

- Syntax
  - Propositions, e.g. “it is raining”
  - Connectives: and, or, not, implies, iff (equivalent)

$$\wedge \quad \vee \quad \neg \quad \longrightarrow \quad \longleftrightarrow$$

  - Brackets, T (true) and F (false)
- Semantics (Classical Boolean)
  - Define how connectives affect truth
    - “P and Q” is true if and only if P is true and Q is true
  - Use **truth tables** to work out the truth of statements

Propositional Logic

- We can represent real world facts as logical propositions written as *well-formed formulas.* E.g.,
  - It is raining : RAINING
  - It is sunny : SUNNY
  - It is windy : WINDY
  - It is raining then it is not sunny (This is logical conclusion)

RAINING $\rightarrow \neg$ SUNNY (Propositional logic representation)

Propositional Logic

- Propositional logic is the simplest way of attempting representing knowledge in logic using symbols.
  - Symbols represent facts: P, Q, etc..
  - These are joined by logical connectives (and, or, implication) e.g., $P \land Q;\ Q \Rightarrow R$
  - Given some statements in the logic we can deduce new facts (e.g., from above deduce R)

Propositional Logic

- Propositional logic isn't powerful enough as a general knowledge representation language.

- Impossible to make general statements. E.g., "all students sit exams" or "if any student sits an exam they either pass or fail".

- So we need predicate logic.

Predicate Logic

- Propositional logic assumes:
  - There are facts that either hold or do not in the world.
  - Each fact can be in one of two states: true or false.
  - Each fact requires a separate proposition.
- Propositional logic combines atoms
  - An atom contains no propositional connectives
  - Have no structure (today_is_wet, john_likes_apples)
- **Predicates** allow us to talk about
  - Objects: people, house
  - Properties: is_wet(today)
  - Relations: likes(john, apples)
  - True or false
- In predicate logic each atom is a predicate
  - e.g. first order logic, higher-order logic
  - More expressive logic than propositional

Predicate Logic: First order Logic

- First-order logic can also be used to make statements about all the objects in the universe, eg all men are mortal.
- **Co**nstants are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): $\forall X.$ likes(X, apples)
  - Exists at least one object (Existential): $\exists X.$ likes(X, apples)

First-Order Logic (FOL)

- A key element of FOL are *predicates*, which are used to describe objects, properties, and relationships between objects

  E.g. $On(x,y)$
- A *quantified statement* is a statement that applies to a class of objects

  E.g. $\forall x\ On(x, Table) \rightarrow Fruit(x)$
  - This means that there is only fruit on the table
  - The first element is called a *quantifier*, x is a *variable* and *Table* is a *constant*
  - *On* is a *predicate*
- The use of *quantifiers* allows FOL to handle *infinite domains*, while propositional logic can only handle finite domains.

Universal Quantification

- Syntax: $\forall$*variables sentence*
- E.g. Everyone taking AI is smart.

  $\forall x\ Taking(x, AI) \rightarrow Smart(x)$
- Semantics: $\forall x\ S$ is equivalent to the *conjunction of instantiations* of S:

$$
\begin{aligned}
& Taking(John, AI) \rightarrow Smart(John) \\
\wedge\ & Taking(Ann, AI) \rightarrow Smart(Ann) \\
\wedge\ & \ldots
\end{aligned}
$$

- Typically, $\rightarrow$ is the main connective with $\forall$.

Universal Quantification: Example

- What does this statement mean:

  $\forall x\ Taking(x, AI) \land Smart(x)$

Existential Quantification

- Syntax: $\exists$*variables sentence*
- Someone taking AI is smart:

  $\exists x\ Taking(x, AI) \wedge Smart(x)$
- Semantics: $\exists x\ S$ is equivalent to the *disjunction of instantiations* of S

$$
\begin{aligned}
& (Taking(Ann, AI) \wedge Smart(Ann)) \\
\vee\ & (Taking(John, AI) \wedge Smart(John)) \\
\vee\ & \ldots
\end{aligned}
$$

- Typically, $\wedge$ is the main connective with $\exists$.

Example

- What does this mean:

  $\exists x\ Taking(x,AI) \rightarrow Smart(x)$

Predicate Logic

- In predicate logic the basic unit is a predicate/ argument structure called an atomic sentence:
  - likes(alison, chocolate)
  - tall(fred)
- Arguments can be any of:
  - constant symbol, such as 'alison'
  - variable symbol, such as X
  - function expression, e.g., motherof(fred)
- So we can have:
  - likes(X, richard)
  - friends(motherof(joe), motherof(jim))

Predicate logic: Syntax

- These atomic sentences can be combined using logic connectives
  - likes(john, mary) $\wedge$ tall(mary)
  - tall(john) $\Rightarrow$ nice(john)
- Sentences can also be formed using quantifiers $\forall$ (for all) and $\exists$ (there exists) to indicate how to treat variables:
  - $\forall$ X lovely(X)   Everything is lovely.
  - $\exists$ X lovely(X)   Something is lovely.
  - $\forall$ X in(X, garden) $\Rightarrow$lovely(X)   Everything in the garden is lovely.

Predicate Logic: Syntax

- Can have several quantifiers, e.g.,
  - $\forall$ X $\exists$ Y loves(X, Y)
  - $\forall$ X handsome(X) $\Rightarrow$ $\exists$ Y loves(Y, X)
- So we can represent things like:
  - All men are mortal.
  - No one likes brussels sprouts.
  - Everyone taking AI will pass their exams.
  - Every race has a winner.
  - John likes everyone who is tall.
  - John doesn't like anyone who likes brussel sprouts.
  - There is something small and slimy on the table.

Predicate Logic: Semantics

- There is a precise meaning to expressions in predicate logic.
- Like in propositional logic, it is all about determining whether something is true or false.
- $\forall$ X P(X) means that P(X) must be true for every object X in the domain of interest

Predicate Logic: Semantics

- $\exists$ X P(X) means that P(X) must be true for at least one object X in the domain of interest.

- So if we have a domain of interest consisting of just two people, john and mary, and we know that `tall(mary)` and `tall(john)` are true, we can say that $\forall$ `X tall(X)` is true.

Proof and inference

- Again we can define inference rules allowing us to say that if certain things are true, certain other things are sure to be true, e.g.
- $\forall$ X P(X) $\Rightarrow$ Q(X)
  P(something)
  ----------------- (so we can conclude)
  Q(something)
- This involves matching P(X) against P(something) and binding the variable X to the symbol something.

Proof and Inference

- What can we conclude from the following?
  - $\forall X \ \ \text{tall}(X) \Rightarrow \text{strong}(X)$
  - tall(john)
  - $\forall Y \ \ \text{strong}(Y) \Rightarrow \text{loves}(\text{mary}, Y)$

Prolog and Logic

- The language which is based upon predicate logic is PROLOG.
- But it has slightly difference in syntax.
  - a(X) :- b(X), c(X). *Equivalent to*
  - $\forall$ X a(X) $\Leftarrow$ b(X) $\wedge$ c(X) ***Or equivalently***
  - $\forall$ X b(X) $\wedge$ c(X) $\Rightarrow$ a(X)
- Prolog has a built in proof/inference procedure, that lets you determine what is true given some initial set of facts. Proof method called "RESOLUTION".

Other Logics

- Predicate logic not powerful enough to represent and reason on things like time, beliefs, possibility.
  - "He may do X"
  - He will do X.
  - I believe he should do X.
- Specialised logics exist to support reasoning on this kind of knowledge

Motivation

- The major motivation for choosing logic as representation tool is that we can reason with that knowledge.