CS249
Artificial Intelligence

Week 10

Slides prepared by:
2201AI48
2201AI49
2201AI50
2201AI51
2201AI52

Limitations of Propositional Logic

- Propositional logic has simple syntax and semantics, hence, very limited expressive power.
- It only has one representational device, the proposition, and cannot generalize
  - Input: facts; Output: facts
  - Result: Many, many rules are necessary to represent any non trivial world
  - It is impractical for even very small worlds

First Order Logic (FOL)

- Also known as Predicate Logic or First Order Predicate Logic
- It is a robust technique to represent objects as well as their relationships.
- Sufficiently expressive to represent natural language statements in a concise way
- According to it the world contains objects, relations, and functions.

1. Objects:
Denote any real-world entity or any variable. E.g., A, B, theories, circles etc.

2. Relations:
Relations represent the links between different objects. Relations can be unary(relations defined for a single term) and n-ary(relations defined for n terms). E.g., blue, round (unary); friends, siblings (binary); etc.

3. Functions:
Functions map their input object to the output object using their underlying relation. Eg: father_of(), mother_of() etc.

Cont.

- Predicates are propositions consisting of expression of one or more variables determined on some specific domain.
- It is an extension to propositional logic in which quantifiers can bind variables in sentences
- For example :
  - $\forall$ x Man(x) $\Longrightarrow$Mortal(x)
  - Man(Tom)
  - What can we infer from these?
  - Can infer Mortal(Tom).

Syntax of FOL

- Syntax represents the rules to write expressions in FOL.
- Basic Elements of FOL :

Element	Example	Meaning
Constant	1, 2, A, John, Mumbai, cat, ...	Values that can not be changed
Variables	x, y, z, a, b, ...	Can take up any value and can also change
Predicates	Brother, Father, >, ...	Defines a relationship between its input terms
Function	sqrt, LeftLegOf, ...	Computes a defined relation of input term
Connectives	$\land$, $\lor$, $\neg$, $\Rightarrow$, $\Leftrightarrow$	Used to form complex sentences using atomic sentences
Equality	==	Relational operator that checks equality
Quantifier	$\forall$, $\exists$	Imposes a quantity on the respective variable

Cont.

- Symbols - The basic syntactic elements of first-order logic that stand for objects, relations, and functions. Types : constant symbols, which stand for objects; predicate symbols, which stand for relations; and function symbols, which stand for functions.
- Interpretation - specifies exactly which objects, relations and functions are referred to by the constant, predicate, and function symbols.
- Terms – A term is a logical expression that refers to an object
- Atomic sentence (or atom for short) is formed from a predicate symbol optionally followed by a parenthesized list of terms, such as Brother(Richard, John).
- Complex sentences can be constructed by combining atomic sentences using connectives like AND (∧), OR (∨), NOT (¬), IMPLIES (⇒), IF AND ONLY IF (⇔) etc. For, connectives c1, c2, etc. we can have this representation:
Predicate 1(term 1, term 2,...)c1 Predicate 2(term 1, term 2,...)c2...

Truth in FOL

- An atomic sentence predicate($term_1$,…, $term_n$) is true iff the objects referred to by $term_1$,…, $term_n$ are in the relation referred to by predicate.
- Example :
  - ● Object symbols
  - – Richard → Richard the Lionheart
  - – John → the evil King John
  - ● Predicate symbol
  - – Brother →the brotherhood relation
  - ● Atomic sentence
  - – Brother(Richard, John)

  true iff Richard the Lionheart and the evil King John are in the brotherhood relation in the model

Quantifiers

- Allows statements about entire collections of objects rather than having to enumerate the objects by name

- **Universal quantifier**: $\forall x$
  Asserts that a sentence is true for all values of variable x
  Def: The phrase “for every”(or its equivalents) is called **universal quantifier**.
  $\forall x$ Loves(x, FOPC)
  $\forall x$ Whale(x) $\Rightarrow$ Mammal(x)
  $\forall x$ Grackles(x) $\Rightarrow$ Black(x)
  $\forall x$ ($\forall y$ Dog(y) $\Rightarrow$ Loves(x,y)) $\Rightarrow$ ($\forall z$ Cat(z) $\Rightarrow$ Hates(x,z))

- **Existential quantifier**: ∃

  Def: The phrase “there exists”(or its equivalents) is called **existential quantifier.**

  Asserts that a sentence is true for at least one value of a variable x

  $\exists x \; Loves(x, FOPC)$

  $\exists x(Cat(x) \land Color(x,Black) \land Owns(Mary,x))$

  $\exists x(\forall y \; Dog(y) \Rightarrow Loves(x,y)) \land (\forall z \; Cat(z) \Rightarrow Hates(x,z))$

Use of Quantifiers

- Universal quantification naturally uses implication:
  $\forall x$ Whale(x) $\wedge$ Mammal(x)
  Says that everything in the universe is both a whale and a mammal.

- Existential quantification naturally uses conjunction:
  $\exists x$ Owns(Mary,x) $\Rightarrow$ Cat(x)
  Says either there is something in the universe that Mary does not own or there exists a cat in the universe.
  $\forall x$ Owns(Mary,x) $\Rightarrow$ Cat(x)

- Says all Mary owns is cats (i.e. everthing Mary owns is a cat). Also true if Mary owns nothing.
  $\forall x$ Cat(x) $\Rightarrow$ Owns(Mary,x)
  Says that Mary owns all the cats in the universe.
  Also true if there are no cats in the universe

Negating Quantifiers

What happens when we negate an expression with quantifiers?

What does our intuition say?

Original	Negation
Every positive integer is prime	There is a positive integer that is not prime.
$\forall x \; \texttt{Prime}(x)$	$\exists x(\neg \; \texttt{Prime}(x))$
Domain of discourse: positive integers	Domain of discourse: positive integers

For a existential quantifier...

Original

There is a positive integer which is prime and even.

$\exists x(\texttt{Prime}\ x \land \texttt{Even}\ x)$

Domain of discourse: positive integers

Negation

Every positive integer is composite or odd.

$\forall x(\neg\texttt{Prime}\ x \lor \neg\texttt{Even}\ x)$

Domain of discourse: positive integers

To negate an expression with a quantifier

1. Switch the quantifier ($\forall$ becomes $\exists$, $\exists$ becomes $\forall$)
2. Negate the expression inside

- We can think of these negations as applications of DeMorgan’s Laws. Let your domain of discourse be the set containing $d1$, $d2$, ... , dn.
  $\exists x(P\ x\ )$ is equivalent to $P\ d1 \vee P\ d2 \vee \cdots \vee P(dn)$
  $\forall x(P\ x\ )$ is equivalent to $P\ d1 \wedge P\ d2 \wedge \cdots \wedge P(dn)$

  Since negating flips ANDs with ORs, it also flips $\exists$ with $\forall$.

Let’s try some examples:

Negate these sentences in English and translate the original and negation to predicate logic.

1. All cats have nine lives. $\forall x\ Cat\ x \rightarrow NumLives\ x$, 9
   Neg: $\exists x(Cat\ x \wedge \neg\ NumLives\ x$, 9) “There is a cat without 9 lives.”

2. All dogs love every person. $\forall x \forall y\ Dog\ x \wedge Human(y) \rightarrow Love\ x, y$
   Neg: $\exists x \exists y(Dog\ x \wedge Human\ y \wedge \neg Love\ x, y$)
   “There is a dog who does not love someone”.
   (or)
   “There is a dog and a person such that the dog doesn’t love that person.”

3. There is a cat that loves someone.

$\exists x \exists y(Cat\ x \land Human\ y \land Love(x, y))$

Neg: $\forall x \forall y([Cat\ x \land Human\ y\ ] \rightarrow \neg Love\ x, y\ )$

"For every cat and every human, the cat does not love that human."

(or)

"Every cat does not love any human" ("no cat loves any human")

- **Interpretation**:
  - In first-order logic, an interpretation assigns meaning to the symbols, predicates, functions, and variables used in logical formulas.
  - An interpretation consists of:
    - A domain of discourse: This is a set of objects over which the variables of the logical language range.
    - Assignments of meanings to predicates: Each predicate symbol is interpreted as a relation over the domain. For example, if we have a predicate symbol "P(x)" representing "x is a person," an interpretation might assign it the meaning "x is older than y."
    - Assignments of meanings to function symbols: Each function symbol is interpreted as a function from the domain to itself or to another subset of the domain. For instance, a function symbol "f(x)" might be interpreted as "x squared."
    - Assignments of meanings to constants: Constants are interpreted as specific elements of the domain.
  - An interpretation allows us to evaluate the truth of logical formulas based on the assigned meanings.

- **Validity**:
  - A formula in first-order logic is considered valid if it holds true in every interpretation, regardless of the specific meanings assigned.
  - For example, the formula $\forall x\, (P(x) \rightarrow P(x))$ ("For all x, if x satisfies property P, then x satisfies property P") is valid because it is true in every interpretation. It reflects a logical truth, stating that anything that satisfies property P also satisfies property P.

- **Satisfiability**:
  - A formula is satisfiable if there exists at least one interpretation under which it evaluates to true.
  - For example, the formula $\exists x\, (P(x) \land Q(x))$ ("There exists an x such that x satisfies both properties P and Q") is satisfiable if there is at least one element in the domain that satisfies both P and Q.
  - If a formula is not satisfiable, it means that no matter how we assign meanings to its symbols, there is no interpretation under which the formula can be true.

- Interpretations, validity, and satisfiability are fundamental concepts in first-order logic that enable us to reason about the truth or falsehood of logical statements and make inferences based on logical rules. They form the basis for understanding the semantics and properties of logical formulas within the framework of first-order logic.

Inference in first-order logic:

- Inference in first-order logic involves the process of drawing conclusions from a set of premises using logical rules. It's essentially about determining whether one statement logically follows from others. Let's break down the process:

1. **Premises**:
    1. In first-order logic, premises are statements or formulas that are assumed to be true. They serve as the starting point for making inferences.
    2. Premises can be atomic formulas (statements without connectives), such as "P(x)" or "Q(y)", or more complex formulas built from atomic formulas using logical connectives (∧, ∨, →, ↔), quantifiers (∀, ∃), and variables.
2. **Logical Rules**:
    1. Inference in first-order logic relies on logical rules that govern the manipulation and deduction of logical formulas.
    2. These rules include:
        1. Modus Ponens: If we have the premises "P → Q" and "P", then we can infer "Q".
        2. Universal Instantiation: If we have the premise "∀x P(x)", we can infer "P(a)" for any individual constant "a".
        3. Existential Instantiation: If we have the premise "∃x P(x)", we can infer "P(a)" for some individual constant "a".
        4. Generalization: If we have the premise "P(a)" for some individual constant "a", we can infer "∀x P(x)".
        5. Existential Generalization: If we have the premise "P(a)" for some individual constant "a", we can infer "∃x P(x)".
        6. And other rules derived from the logical connectives and quantifiers.

- **Inference Process**:
  - To make an inference, we start with the premises and apply logical rules step by step to derive new conclusions.
  - Each step of the inference process must be justified by applying a valid logical rule.
  - The goal is to derive a conclusion that logically follows from the premises.
- **Validity of Inferences**:
  - An inference is considered valid if the conclusion logically follows from the premises.
  - A valid inference guarantees that if the premises are true, then the conclusion must also be true.
  - Validity is assessed based on the logical rules applied and the properties of the logical formulas involved.
- **Soundness**:
  - A valid inference is sound if the premises are true in some interpretation.
  - Soundness ensures that not only is the inference logically valid, but it also reflects true statements about the domain of discourse.

Propositional logic

- Propositional logic, also known as sentential logic or statement logic, is a branch of symbolic logic that deals with propositions, which are statements that can either be true or false, but not both.
- In propositional logic, these propositions are represented by symbols, typically capital letters like PP, QQ, RR, etc. Logical connectives, such as AND ($\land$), OR ($\lor$), NOT ($\neg$), IMPLIES ($\rightarrow$), and IF AND ONLY IF ($\leftrightarrow$), are used to form compound propositions from simpler ones.

Propositional logic Example

- Statement 1 : If I am the Director then I am well known.
- Statement 2 : I am the Director ,So I am well known.

Coding :
Variable : a) I am the Director
b) I am well known
Coding sentences : 1) $a \rightarrow b$
2) a
3) b

a	**b**	**$a \rightarrow b$**	**$(a \rightarrow b) \land a$**	**$((a \rightarrow b) \land a) \rightarrow b$**
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

- If I am the director then I am will be known , I am not the director , So I am not well known
  - Implication and contrapositive are equivalent
  - Converse and Inverse are equivalent

  $(B \rightarrow A) \equiv \neg A \rightarrow \neg B$

a	b	a → b	(a→b) Λ ¬a	((a→b) Λ ¬a)→b
T	F	T	F	T
F	F	T	T	T
T	F	F	F	T
F	T	T	T	F

Predicate logic Example

- Statement : All child loves its mother

Mother(x,y) → y is mother of x
Love(x,y) → x loves y

$\forall$ x $\exists$ y mother(x,y) $\wedge$ loves(x,y)

Predicate logic Example

- Statement : Wherever Mary goes , so does the lamb . Mary goes to school . So, the Lamb goes to school

Predicate : Goes(x,y) represents y goes to x ,
Goes(Mary,School)

F1 : $\forall$ x (Goes(Mary,x) $\rightarrow$ Goes(Lamb,x)
F2 : Goes(Mary,School)
F3 : (F1 $\wedge$ F2) $\rightarrow$ A is always true

Validity : P is true Under all interpretation
Satisfiability : P is true , under at least one interpretation

Predicate logic Example

- Statement : Not all students take both History and Biology

Student(x) Ξ x is a student
Take(x,y) Ξ subject x is taken by y

$\neg [\forall x \; student(x) \rightarrow Take(History) \land Take(Biology)]$

Synonymous to previous statement
There Exists a student such that he does not take history or does not take biology

$\exists x \; student(x) \land [\neg Take(Hist,x) \lor \neg Take(bio,x)]$

Predicate logic Example

- Statement : Only One student Failed in both History and Biology

Student(x) Ξ x is a student
Fail(x,y) Ξ student y failed in x

$\exists$ x [(student(x) $\wedge$[Fail(Hist,x) $\wedge$ Fail(bio,x)]) $\mathbf{\wedge}$
($\forall$ y [($\neg$(x=y) $\wedge$student(y)) $\rightarrow$ [$\neg$ Fail(Hist,x) $\vee$ $\neg$ Fail(bio,x)]])

Predicate logic Example

- Statement : The best score in History is better than the best score in Biology

Score(Subject , Student)
Greater(x,y) : x>y

$\forall$ x [(student(x) $\wedge$Take(Bio,x) $\rightarrow$ $\exists$ y [Student(y) $\wedge$ Take(Hist,y) $\wedge$ Greater(Score(Hist,y) , Score(Bio,y)]]

Russel’s Paradox

- There is a Single Barber in the Town
- Those and only those who don’t shave themselves are shaved by others
- Who shaved the Barber

Let B(x) denote "Person x is the barber."
Let S (x , y) denote "Person y shaved person x.“

1. There is a single barber in town:

$$\exists x\, (B(x) \land \forall y\, (B(y) \rightarrow y = x))$$

2. Those and only those who don’t shave themselves are shaved by others:

$$\forall x\, (\neg S(x , x) \leftrightarrow \exists y\, (B(y) \land S (x, y)))$$

3. "Who shaved the barber?":

$$\exists y\, (B(y) \land S (y, y))$$

Inference

In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, so generating the conclusions from evidence and facts is termed as Inference.

In AI, inference can be categorized into two types: deductive inference and inductive inference. Deductive inference involves reasoning from general principles to specific conclusions, while inductive inference involves inferring general principles or rules based on specific observations or data.

Inference Rules:

Inference rules are the templates for generating valid arguments. Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.In inference rules, the implication among all the connectives plays an important role. Following are some terminologies related to inference rules:

1. **Implication**: It is one of the logical connectives which can be represented as P → Q. It is a Boolean expression.
2. **Converse**: The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as Q → P.
3. **Contrapositive**: The negation of converse is termed as contrapositive, and it can be represented as ¬ Q → ¬ P.
4. **Inverse**: The negation of implication is called inverse. It can be represented as ¬ P → ¬ Q.

Inference in First-Order Logic

Inference in First-Order Logic is used to deduce new facts or sentences from existing sentences. Before understanding the FOL inference rule, let's understand some basic terminologies used in FOL.

1. **Substitution**:Substitution is a fundamental operation performed on terms and formulas. It occurs in all inference systems in first-order logic. The substitution is complex in the presence of quantifiers in FOL. If we write F[a/x], so it refers to substitute a constant "a" in place of variable "x".
2. **Equality**:First-Order logic does not only use predicate and terms for making atomic sentences but also uses another way, which is equality in FOL. For this, we can use equality symbols which specify that the two terms refer to the same object.

   Example: Brother (John) = Smith.

   As in the above example, the object referred by the Brother (John) is similar to the object referred by Smith. The equality symbol can also be used with negation to represent that two terms are not the same objects.

FOL inference rules for quantifier:

1. <u>**Universal Generalization:**</u>
   A. Universal generalization is a valid inference rule which states that if premise P(c) is true for any arbitrary element c in the universe of discourse, then we can have a conclusion as $\forall$ x P(x).
   B. It can be represented as:
   C. This rule can be used if we want to show that every element has a similar property.
   D. In this rule, x must not appear as a free variable.
   E. Example: Let's represent, P(c): "A byte contains 8 bits", so for $\forall$ x P(x) "All bytes contain 8 bits.", it will also be true.

2. Universal Instantiation:

A. Universal instantiation is also called as universal elimination or UI is a valid inference rule. It can be applied multiple times to add new sentences.
B. The new KB is logically equivalent to the previous KB.
C. As per UI, we can infer any sentence obtained by substituting a ground term for the variable.The UI rule state that we can infer any sentence P(c) by substituting a ground term c (a constant within domain x) from $\forall$ x P(x) for any object in the universe of discourse.
D. It can be represented as:

A. Example:1.IF "Every person like ice-cream"=> $\forall$x P(x) so we can infer that "John likes ice-cream" => P(c)

3. Existential Instantiation:

A. Existential instantiation is also called as Existential Elimination, which is a valid inference rule in first-order logic.It can be applied only once to replace the existential sentence.

B. The new KB is not logically equivalent to old KB, but it will be satisfiable if old KB was satisfiable.

C. This rule states that one can infer P(c) from the formula given in the form of ∃x P(x) for a new constant symbol c.The restriction with this rule is that c used in the rule must be a new term for which P(c) is true.

4. Existential introduction

A. An existential introduction is also known as an existential generalization, which is a valid inference rule in first-order logic.

B. This rule states that if there is some element c in the universe of discourse which has a property P, then we can infer that there exists something in the universe which has the property P.

Example: Let's say that,

"Priyanka got good marks in English."

"Therefore, someone got good marks in English."

Reasoning in first order logic

First-order logic statements can be divided into two parts:

Subject: Subject is the main part of the statement.

Predicate: A predicate can be defined as a relation, which binds two atoms together in a statement.

Quantifiers in First-order logic:

A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:

Universal Quantifier, (for all, everyone, everything)

Existential quantifier, (for some, at least one).

Free and Bound Variables: The quantifiers interact with variables which appear in a suitable way. There are two types of variables in First-order logic which are given below:

Free Variable: A variable is said to be a free variable in a formula if it occurs outside the scope of the quantifier.

Bound Variable: A variable is said to be a bound variable in a formula if it occurs within the scope of the quantifier.

Example:

1. The law says that it is a crime for gawl to sell potion formula to hostile nations.
2. The country in Rome, an enemy of gawl has acquired some potion formula and all its formula are sold to it by traitor.
3. Traitor is gawl.
4. Is traitor a criminal?

Further Reading Task

- https://drive.google.com/drive/folders/1jIW3LhPxzsLCxBqdWmC8pTkTzXK7o_5Z?usp=sharing

- https://discrete.openmathbooks.org/dmoi2/sec_propositional.html

Thank you………