Artificial Intelligence

Lec 18: Propositional Logic (contd.), First Order Logic

Pratik Mazumder

Content taken from various sources

Resolution

- Principle
  - Suppose x is a literal, and S1 and S2 are two sets of propositional sentences represented in causal form.

  - If we have 2 clauses $(x \lor S1)$ and $(\neg x \lor S2)$
  - Then we get $S1 \lor S2$

  - Here $S1 \lor S2$ is the resolvent.
  - x is resolved upon.

- Explanation: If we know that either x or S1 is True and we also know that either x is False or S2 is True, then either S1 or S2 will be True.

- Another way to check is to verify if $(x \lor S1) \land (\neg x \lor S2) \Rightarrow (S1 \lor S2)$ is a tautology.

Resolution

- Take any two clauses where one contains some symbol and the other contains the complement of this symbol:

  $P \lor Q \lor R \qquad\qquad \neg Q \lor S \lor T$

- Merge (resolve) them by throwing away this symbol and its complement, to obtain their resolvent clause.

  $P \lor R \lor S \lor T$

- **After resolving** the two clauses, **if there are no symbols left,** then you have derived a **False or empty clause**.

Resolution

- Resolution Rule of Inference

$$\frac{\begin{array}{l} \alpha \vee \beta \\ \neg\alpha \vee \gamma \end{array}}{\beta \vee \gamma}$$

- Examples

$$\frac{\begin{array}{l} A \vee B \\ \neg B \end{array}}{A}$$

(unit resolution)

$$\frac{\begin{array}{l} A \vee B \vee \neg C \vee D \\ \neg A \vee \neg E \vee F \end{array}}{B \vee \neg C \vee D \vee \neg E \vee F}$$

Inference Technique 3: Inference by Resolution Refutation

- **Show KB ⊨ β** by **proving that KB∧¬β is unsatisfiable**, i.e., **deducing False** from KB∧¬□.

- **Proof by Contradiction**

- **Only a single inference rule can be used (**i.e., only **resolution** can be used**)** after converting KB and ¬□ into the required form.

- Procedure
  - **Add negation of the query** sentence β **to** the **KB**.

  - **Convert** all the sentences in KB into **causal form**.

  - **Iteratively apply resolution** to the **clauses (pick two containing complimentary literals)** in KB and **add the resolvent to the KB**.

  - **Continue until no further resolvents can be obtained** or a **null clause** is obtained.

Inference Technique 3: Inference by Resolution Refutation

- <u>KB</u>
  $A \Leftrightarrow (B \lor C)$
  $\neg A$

  <u>Query</u>
  $\neg B$

- <u>New KB</u>
  $A \Leftrightarrow (B \lor C)$ [convert to Causal Form]
  $\neg A$ [already in Causal Form]
  B [already in Causal Form]

- <u>New KB in Causal Form</u>
  $\neg A \lor B \lor C$
  $\neg B \lor A$
  $\neg C \lor A$
  $\neg A$
  B

Convert $A \Leftrightarrow (B \lor C)$ to causal form

$(A \Rightarrow (B \lor C)) \land ((B \lor C) \Rightarrow A)$ [biconditional elimination]

$(\neg A \lor B \lor C) \land (\neg(B \lor C) \lor A)$ [implication elimination]

$(\neg A \lor B \lor C) \land ((\neg B \land \neg C) \lor A)$ [DeMorgan's Law]

$(\neg A \lor B \lor C) \land (\neg B \lor A) \land (\neg C \lor A)$ [Distributing $\lor$ over $\land$]

3 clauses: $\neg A \lor B \lor C$, $\neg B \lor A$, $\neg C \lor A$

Inference Technique 3: Inference by Resolution Refutation

- New KB in Causal Form
  1. ¬A⋁B⋁C
  2. ¬B⋁A
  3. ¬C⋁A
  4. ¬A
  5. B

Step 1: Resolve 2 and 5

6. A

Step 2: Resolve 4 and 6

7: Empty Clause or False

Therefore, KB entails ¬B

Proved by Contradiction

Inference Technique 3: Inference by Resolution Refutation

- KB
  If the triangle is equilateral then it is isoceles.
  If the triangle is isoceles, then two sides AB and AC are equal.
  If AB and AC are equal, then angles B and C are equal
  ABC is an equilateral triangle.

  Query: Angle B and C are equal.

- Propositional Logic

  Let, E = triangle is equilateral, I = triangle is isoceles, S = two sides AB and AC are equal, A = angle B and C are equal

  KB
  $E \Rightarrow I$
  $I \Rightarrow S$
  $S \Rightarrow A$
  E

  Query: A

Inference Technique 3: Inference by Resolution Refutation

<u>New KB</u>

$E \Rightarrow I$ [convert to Causal Form]

$I \Rightarrow S$ [convert to Causal Form]

$S \Rightarrow A$ [convert to Causal Form]

E [already in Causal Form]

$\neg A$ [already in Causal Form]

Conversion to Causal Form

$E \Rightarrow I$ –> $\neg E \lor I$ [Implication Elimination]

$I \Rightarrow S$ –> $\neg I \lor S$ [Implication Elimination]

$S \Rightarrow A$ –> $\neg S \lor A$ [Implication Elimination]

<u>New KB in Causal Form</u>

1. $\neg E \lor I$
2. $\neg I \lor S$
3. $\neg S \lor A$
4. E
5. $\neg A$

Inference Technique 3: Inference by Resolution Refutation

<u>New KB in Causal Form</u>

1. $\neg E \lor I$
2. $\neg I \lor S$
3. $\neg S \lor A$
4. E
5. $\neg A$

Step 1: Resolve 3 and 5

6. $\neg S$

Step 2: Resolve 2 and 6

7. $\neg I$

Step 3: Resolve 1 and 7

8. $\neg E$

Step 4: Resolve 4 and 8

9. Empty Clause or False

Therefore, KB entails A, i.e. angle B and C are equal
Proved by Contradiction

Inference Technique 3: Inference by Resolution Refutation

- Some points
  - If a new clause contains duplicates of the same symbol, delete the duplicates

    $P \lor R \lor P \lor T \equiv \quad P \lor R \lor T$

  - If a clause contains the symbol and its complement, then the clause is a tautology and is useless and can be thrown away without including in the KB, e.g.
    a1. $\neg A \lor B \lor C$
    a2. $\neg B \lor A$

    Resolvent of a1 and a2 is $B \lor C \lor \neg B \equiv T \lor C \equiv T$
    Which is a tautology and can be discarded

  - When the **two clauses contain more than one pair of complementary literals**, the **resolution** rule can be applied to only one pair of complimentary literals at a time, e.g., $A, \neg A$ and $B, \neg B$ above.
    - However, in such cases, the result is always a tautology and thus discarded.

Limitations of Propositional Logic

- We have looked at propositional logic and how reasoning is carried out in Propositional Logic.
- We will now discuss a more powerful/expressive form of logic called First Order Logic or Predicate Logic.
- FOL can capture more variety of sentences in the KB as compared to Propositional Logic.
- Let us consider an example to understand the limitations of Propositional Logic.
- Suppose a Knowledge Base KB has the following sentences

  <u>KB</u>
  All dogs are faithful.
  Tommy is a dog.

  <u>Query</u>
  Tommy is faithful.

Limitations of Propositional Logic

Using Propositional Logic

<u>KB</u>
P = All dogs are faithful.
T = Tommy is a dog.

<u>Query</u>
R = Tommy is faithful.

$P \land T \nvDash R$

- We cannot infer this in propositional logic.

- Even though we as humans can easily deduce this, we have to provide a mechanical method for the computer to logically understand and infer this.

Limitations of Propositional Logic

Using Propositional Logic:

If Tom is a hardworking student and Tom is an intelligent student, then Tom scores high marks.

P = Tom is a hardworking student.
T = Tom is an intelligent student.
S = Tom scores high marks.

$P \land T \Rightarrow S$ = If Tom is a hardworking student and Tom is an intelligent student, then Tom scores high marks.

- So Propositional Logic is working in this case.
- But what if you want to now do the same for another student, say Animesh or Pranay?
- You will have to create separate propositions for Animesh, Pranay, and any one else.
    - Have to explicitly encode the statements for each student separately in PL which is not feasible.
- Suppose after doing this for students, now a new student joins the class, and we have to repeat the process.

Limitations of Propositional Logic

- Sentence: All students who are hardworking and intelligent score high marks.

  If we could write this as

  For all x such that x is a student and x is intelligent and x is hardworking, then x scores high marks.

  x is a variable and can be used to correspond to Praveen, Animesh, Pranay, Rahul, etc.

- Not possible in PL.

- In general, propositional logic can deal with only a finite number of propositions.

- If there are only 3 dogs, Meeka, Arya, Hodor, then we can use PL and write:
  M: Meeka is faithful, A: Arya is faithful, H: Hodor is faithful.

- What if there are an infinite number of propositions, e.g., how to represent “all dogs are faithful” in PL: $M \wedge A \wedge H \wedge \ldots$

First-Order Logic

- FOL or Predicate Logic is a generalization of propositional logic that allows us to express and infer arguments in infinite models like
    - All men are mortal.
    - Some birds cannot fly.
    - Atleast one planet has life on it.
- We can express these sentences in FOL because in addition to the concepts of propositional logic, we have the concept of variables and the concept of quantifiers.

First-Order Logic: Syntax

- Syntax can be defined using:
    - Terms
    - Predicates
    - Quantifiers

First-Order Logic: Term

- Term: denotes some objects other than true or false, e.g.,
  - <u>Tommy</u> is a dog.
  - All <u>men</u> are mortal.
- Terms can be constants or variables.
- A constant of type W is a name that denotes a particular object in set W.
  - Example: 5 is a particular object in a set of natural numbers. Tommy is a particular object in a set of dog names, etc.
- A variable of type W is a name that can denote any element in set W.
  - Example: $x \in N$ denotes a natural number, d denotes the name of a dog, etc.

First-Order Logic: Function

- A **functional term of arity n** takes n objects of type $W_1,W_2,...,W_n$ as inputs and returns an object of type W
    - $f(w_1,w_2,...,w_n)$
    - $\underbrace{\text{plus}}_{\text{Functional term}} \; (\underbrace{\underline{3} \,,\, \underline{4}}_{\text{Constant terms}}) = \underline{7}$

- Let plus be a function that takes two arguments of type Natural Number and returns a Natural Number.

- Valid functional terms:
    - plus(2,3)
    - plus(5, plus(7,3))
    - plus(plus(100,plus(1,6)),plus(3,3))

- Invalid functional terms:
    - plus(0,-1)
    - plus(1.2,3.1)

First-Order Logic: Function

- Functions with variable arguments
    - plus(x,y)
    - plus(x,y,z)

First-Order Logic: Predicates

- Predicates are like functions except that their return type is True or False.
- Examples:
  - gt(x,y) is True if x > y.
  - Here gt is a predicate symbol that takes two arguments of type natural number.
  - gt(3,4) is a valid predicate that returns False.
  - gt(3,-4) is not a valid predicate.

First-Order Logic: Types of Predicates

- A predicate with no variables (similar to a proposition)
    - Tommy is a dog
- A predicate with one variable is called a property.
    - dog(x) is True iff x is a dog.
    - mortal(y) is True iff y is mortal.

First-Order Logic: Formulation of Predicates

- Let P(x,y,...) and Q(x,y,...) be two predicates.
- Then the following are also valid predicates
  - $P \lor Q$
  - $P \land Q$
  - $\neg P$
  - $P \Rightarrow Q$

First-Order Logic: Predicate Examples

- If x <u>is a man</u> then x <u>is mortal</u>
    - man(x) ⇒ mortal(x)
- If n <u>is a natural number</u>, then either n <u>is even</u> or n <u>is odd</u>.
    - natural(n) ⇒ (even(n) ∨ odd(n))

First-Order Logic: Quantifiers

- There are two basic quantifiers in FOL:
  - $\forall$ "For all" - Universal quantifier.
  - $\exists$ "There exist" - Existential quantifier.
- Suppose you have a set W containing multiple elements, and x is a variable that can refer to any element in the set W (set of dogs, set humans, etc.).
- $\forall x\, P(x)$ - means that $P(x)$ is True for elements x in the set W.
- $\exists x\, Q(x)$ - means there should be atleast one element in W for which Q is True.
- Suppose the set W refers to days of the week and $Q(x)$ corresponds to x is a holiday.
  - Then we can write $\exists x\, Q(x)$, since only for some x, $Q(x)$ is True.
- Suppose $P(x)$ corresponds to x "ends with" '-day'.
  - Then we can write $\forall x\, P(x)$, since all days of the week end with -day (Mon**day**, Tues**day**, …)

First-Order Logic: Universal Quantifiers

- All dogs are faithful

  faithful(x): x is faithful
  dog(x): x is a dog

  $\forall x\ (dog(x) \Rightarrow faithful(x))$

- Not all birds can fly.

  bird(x): x is a bird
  fly(x): x can fly

- $\neg(\forall x\ (bird(x) \Rightarrow fly(x)))$

- $\exists x\ bird(x)\ \land\ \neg fly(x)$

First-Order Logic: Universal Quantifiers

- Mammals drink milk

- Man is mortal

- Man is a mammal

- Tom is a man

First-Order Logic: Universal Quantifiers

- Mammals drink milk.
  - $\forall x\ (mammal(x) \Rightarrow drink(x, Milk))$
- Man is mortal.
  - $\forall x\ (man(x) \Rightarrow mortal(x))$
- Man is a mammal.
  - $\forall x\ (man(x) \Rightarrow mammal(x))$
- Tom is a man.
  - $man(Tom)$