Artificial Intelligence

Lec 17: Propositional Logic (contd.)

Pratik Mazumder

Content taken from various sources

De Morgan’s Theorem

$\neg(P \land Q) = \neg P \lor \neg Q$

$\neg(P \lor Q) = \neg P \land \neg Q$

P	Q	$P \lor Q$	$\neg(P \lor Q)$	$\neg P$	$\neg Q$	$\neg P \land \neg Q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Equivalences

- Prove that the following formulas are equivalent:
  - $\neg(P \lor (\neg P \land Q))$ and $\neg P \land \neg Q$
  - $\neg(P \Rightarrow Q)$ and $P \land \neg Q$

Variations of Implications

- **Converse** of an implication $P \Rightarrow Q$ is $Q \Rightarrow P$
- **Inverse** of an implication $P \Rightarrow Q$ is $\neg P \Rightarrow \neg Q$
- **Contrapositive** of an implication $\neg Q \Rightarrow \neg P$
- Compare them using Truth Tables and also using the equivalences

Reasoning

- Using the given propositions try to derive new propositions

  If it is the month of July then it rains
  It is the month of July

  Conclude: It rains

  P: It is the month of July
  Q: It rains
  R: $P \Rightarrow Q$

  $$\frac{\begin{array}{l} P \Rightarrow Q \\ P \end{array}}{Q}$$

  Modus Ponens inference rule

Reasoning

- $P \Rightarrow Q$
  P
  ———
  Q

- Prove the Modus Ponens inference rule = Prove $(P \Rightarrow Q) \wedge P \Rightarrow Q$ is a tautology.

Inference Rules

- $$\frac{P \land Q}{P}$$ [And Elimination] [Explanation: If P and Q both are true, then P will also be True]
- $$\frac{P \quad Q}{P \land Q}$$ [And Introduction] [Explanation: If we know that P is True and Q is True, then P and Q will also be True]
- $$\frac{P}{P \lor Q}$$ [Or Introduction] [Explanation: If P is true, then $P \lor Q$ will also be True]
- $$\frac{P \lor Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R}$$ [Or Elimination] [Explanation: If P or Q is true, and $P \Rightarrow R$ and $Q \Rightarrow R$ are True, then R will also be True]

Inference Rules

- $P \Rightarrow Q$ [Chain Rule]
  $Q \Rightarrow R$
  $$\frac{P \Rightarrow Q \quad Q \Rightarrow R}{P \Rightarrow R}$$

- $P \Rightarrow Q$ [Modus Tollens]
  $\neg Q$
  $$\frac{P \Rightarrow Q \quad \neg Q}{\neg P}$$

The **name** of the inference rule used **MUST** be mentioned.

Otherwise No Considerations

Explanations cannot be in place of the name of the inference rules.

Satisfiability

- A sentence is **satisfiable under an interpretation** if that sentence evaluates to **True under that interpretation**.
- If **no interpretation makes the sentence True,** then it is **unsatisfiable.**
- e.g., $P \land \neg P$

Entailment

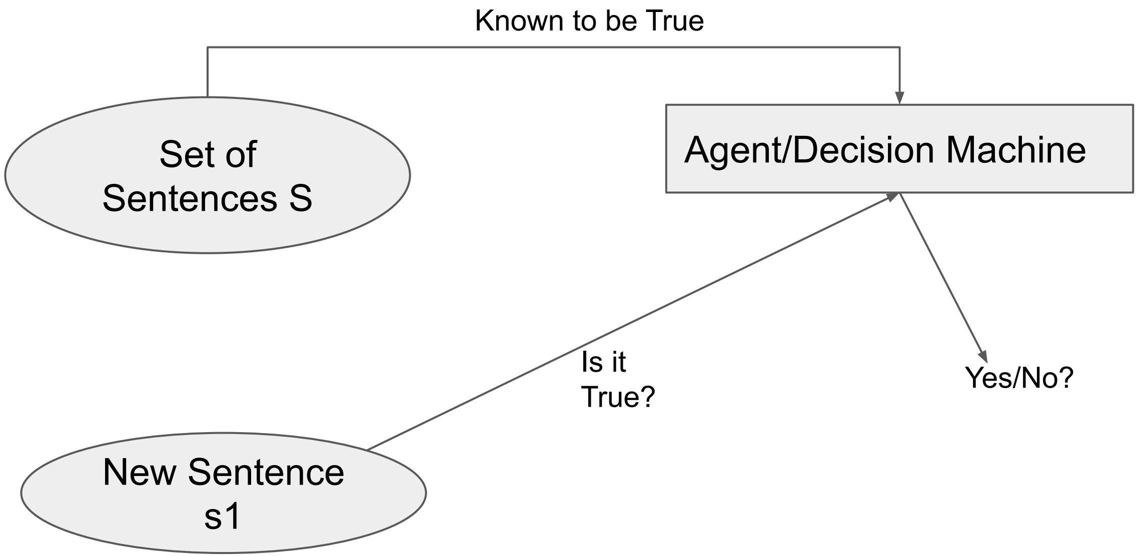
- If we have a **set of sentences S** and a **set of Interpretations I** for which **all sentences in S is True.**
  - If **another sentence s1** is **also true for all the Interpretations in I** then we say that **S entails s1.**
  - This is **logical entailment**.

- S $\models$ s1
- **s1 logically follows S.**
- **s1 is a logical consequence of S.**
- **S logically entails s1.**

- Basically, S $\models$ s1 means that in **every interpretation for which S is True, s1 is also True**.

- **S $\models$ s1 if and only if S $\Rightarrow$ s1 is valid**

Entailment



We can directly check if S entails s1. If yes, then s1 will be True.

Inference Technique 1: Inference by enumeration

- Also called Truth Table Enumeration
- Let: Knowledge Base (KB) = $A \lor C$, $B \lor \neg C$
- New proposition/query $\beta = A \lor B$
- Show that: KB $\models \beta$
- All logically distinct cases must be checked to prove that a sentence can be derived from KB.

Inference Technique 1: Inference by enumeration

- $KB = A \lor C, B \lor \neg C$, $\beta = A \lor B$, Show that: $KB \vDash \beta$
  $\boldsymbol{\alpha} = (A \lor C) \land (B \lor \neg C)$

A	**B**	**C**	**A∨C**	**B∨¬C**	**α**	**A∨B**	**α⇒□**
T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T
T	F	T	T	F	F	T	T
T	F	F	T	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	T	F	T	T
F	F	T	T	F	F	F	T
F	F	F	F	T	F	F	T

Inference Technique 1: Inference by enumeration

- KB = A⋁C, B⋁¬C, β = A⋁B, Show that: KB ⊨ β
  α = (A⋁C)⋀(B⋁¬C)

Interpretations for which all sentences in KB are True

A	B	C	A⋁C	B⋁¬C	α	A⋁B	α⇒□
T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T
T	F	T	T	F	F	T	T
T	F	F	T	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	T	F	T	T
F	F	T	T	F	F	F	T
F	F	F	F	T	F	F	T

Inference Technique 1: Inference by enumeration

- KB = $A \lor C$, $B \lor \neg C$, β = $A \lor B$, Show that: KB ⊨ β
  α = $(A \lor C) \land (B \lor \neg C)$

Interpretations for which all sentences in KB are True

A	B	C	$A \lor C$	$B \lor \neg C$	α	$A \lor B$	α⇒□
T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T
T	F	T	T	F	F	T	T
T	F	F	T	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	T	F	T	T
F	F	T	T	F	F	F	T
F	F	F	F	T	F	F	T

□ is True for all interpretations for which all sentences in KB are True. So KB ⊨ □ is valid

Inference Technique 1: Inference by enumeration

- KB = $A \lor C$, $B \lor \neg C$, $\beta = A \lor B$, Show that: KB $\vDash \beta$
  $\alpha = (A \lor C) \land (B \lor \neg C)$

A	B	C	$A \lor C$	$B \lor \neg C$	α	$A \lor B$	$\alpha \Rightarrow$ □
T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T
T	F	T	T	F	F	T	T
T	F	F	T	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	T	F	T	T
F	F	T	T	F	F	F	T
F	F	F	F	T	F	F	T

Checking for entailment

Inference Technique 1: Inference by enumeration - Remarks

- Inference by enumeration requires building a complete truth table in order to determine if a sentence is entailed by the knowledge base.

- If 100 atomic propositions are present, then the truth table will have 2^{100} rows.

- Therefore, inference by enumeration is very slow and takes exponential time.

Inference Technique 2: Natural Deduction using Sound Inference Rules

- A more **efficient** algorithm than enumeration that uses a set of **inference rules** and logical **equivalences** to
    - **incrementally deduce new sentences** that are true **given the initial set of sentences** in the knowledge base (KB).

- Requires constructing a proof.

- A **proof** is a **sequence of inference steps** that leads **from KB to β** (query).

  KB:
  $(P \wedge Q) \Rightarrow R$
  $(S \wedge T) \Rightarrow Q$
  S
  T
  P

  Query: R

Proof by Natural Deduction

KB:
$(P \wedge Q) \Rightarrow R$
$(S \wedge T) \Rightarrow Q$
S
T
P

Query: R

1.	S	[Premise, i.e., given sentence in KB]
2.	T	[Premise]
3.	P	[Premise]
4.	$S \wedge T$	[And Introduction to 1,2]
5.	$(S \wedge T) \Rightarrow Q$	[Premise]
6.	Q	[Modus Ponens to 4,5]
7.	$P \wedge Q$	[And Introduction to 3,6]
8.	$(P \wedge Q) \Rightarrow R$	[Premise]
9.	R	[Modus Ponens to 8,9]

Proved…

Monotonicity Property

- Natural Deduction relies on the **monotonicity property** of Propositional Logic, i.e.,
    - **Deriving** a new **sentence** and **adding** it to the **knowledge base does not affect** what can be entailed from the **original KB.**
- Hence, we can incrementally add new True sentences that are derived in any order.
- Once something is proved True, it will remain True.

Clause

- Literal: A single proposition or its negation

  $P, \neg P$

- A clause is a disjunction of literals

  $P \lor Q \lor \neg R$

- Clausal form is very useful for inferencing

Converting a Compound Proposition to Clausal Form

Consider the sentence

$\neg(A \Rightarrow B) \vee (C \Rightarrow A)$

1. Eliminate the implication sign

   $\neg(\neg A \vee B) \vee (\neg C \vee A)$

2. Eliminate double negation and reduce the scope of “not” signs (De-Morgan’s law).

   $(A \wedge \neg B) \vee (\neg C \vee A)$

3. Convert to conjunctive normal form by using distributive and associative laws.

   $(A \vee \neg C \vee A) \wedge (\neg B \vee \neg C \vee A)$

   $(A \vee \neg C) \wedge (\neg B \vee \neg C \vee A)$

4. Get the set of clauses

   $(A \vee \neg C)$

   $(\neg B \vee \neg C \vee A)$

Clausal Form allows us to apply Inference by Resolution.

Converting to Clausal Form: Examples

implication elimination

$A \Rightarrow B$ —> $\neg A \vee B$ [one clause]

$A \Leftrightarrow B$ —> $(A \Rightarrow B) \wedge (B \Rightarrow A)$ —> $(\neg A \vee B) \wedge (\neg B \vee A)$ —> two clauses: $\neg A \vee B$, $\neg B \vee A$

DeMorgan's Law

$\neg(A \vee B)$ —> $\neg A \wedge \neg B$ —> two clauses: $\neg A$, $\neg B$

$\neg(A \wedge B)$ —> $\neg A \vee \neg B$ [one clause]

$(A \wedge B) \wedge C$ —> $A \wedge B \wedge C$ —> three clauses: A, B, C

$(A \vee B) \vee C$ —> $A \vee B \vee C$ [one clauses]

distributing $\vee$ over $\wedge$

$(A \wedge B) \vee C$ —> $(A \vee C) \wedge (B \vee C)$ —> two clauses: $A \vee C$, $B \vee C$

$(A \vee B) \wedge C$ —> two clauses: $A \vee B$, C

Always mention the equivalences/laws being used - distributing.. or implication elimination or …..