# Artificial Intelligence
## AI 2002

## Lecture 13

Mahzaib Younas

Lecturer Department of Computer Science

FAST NUCES CFD

$$\alpha \models \beta \text{ if and only if } M(\alpha) \subseteq M(\beta)$$

# Logic --- Valid

▸ A sentence is **valid** if it is *true* in *all models (interpretations).*

▸ **Valid Sentances:**

true, ¬ false, P ∨ ¬P

▸ Valid sentences are also known as **tautologies**—they are *necessarily true.*

▸ From the definition of entailment, we can derive the **deduction theorem,** for example

*For any sentences $\alpha$ and $\beta$, $\alpha \models \beta$ if and only if the sentence $(\alpha \Rightarrow \beta)$ is valid.*

# Logic --- Satisfiability

▸ A sentence is **satisfiable** if it is *true* in, or satisfied by, *some* model.

▸ A sentence is **satisfiable** if and only if it's *true* in at least *one interpretation*.

▸ Satisfiable Sentence:
  ◦ true, P,¬P

▸ A sentence is **unsatisfiable** if and only if it's truth value is *false* in *all interpretation*.
  ◦ ¬ true, false, P ∧ ¬P

# Examples

| P | Q | ¬P | P∧Q | P∨Q | P→Q | Q→P | P↔Q |
|---|---|---|---|---|---|---|---|
| f | f | t | f | f | t | t | t |
| f | t | t | f | t | t | f | f |
| t | f | f | f | t | f | t | f |
| t | t | f | t | t | t | t | t |

| Sentences | Valid/ Satisfiability? | Interpretation |
|---|---|---|
| Smoke → Smoke | Valid | |
| Smoke ∨ ¬Smoke | Valid | |
| Smoke → Fire | Satisfiable | Smoke = **t** Fire = **f,** "smoke implies fire" |
| (s → f) → (¬s → ¬f) | Satisfiable | s = **f**, f = **t** (s e f) = **t**, (¬s e ¬f) = **f** "Smoke implies fire **implies** not smoke implies not fire." |
| (s → f) → (¬f → ¬s) | Valid | Contrapositive "smoke implies fire **implies** not fire implies not smoke" |

# Logic --- Satisfiability

- The problem of satisfiability of a sentence is very **related** to the **constraint satisfaction problem**, that has to **find a legal assignment** that can satisfy the constraint,

- Similarly, we have to *find interpretation in such as way that sentence holds that interpretation*.

- Can be solved the problem using the **brute-force method** of enumerating all possible interpretations.
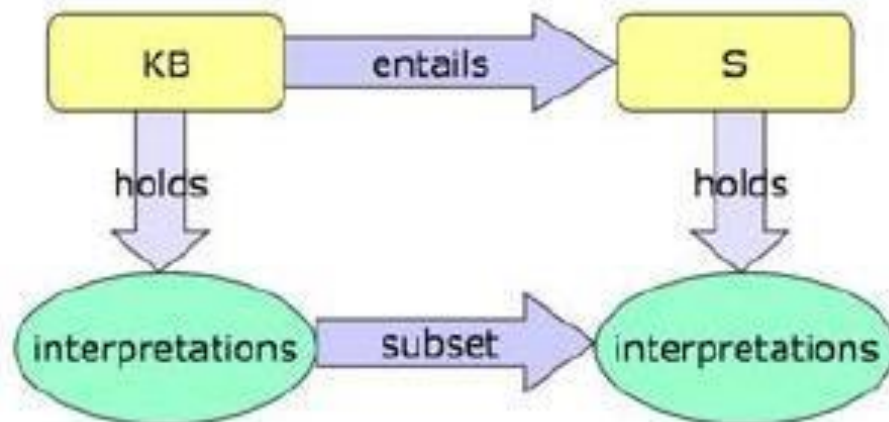
# Entailment

- Entailment that signifies a **relationship** between a knowledge base and another sentence.
- If every interpretation that satisfies the $KB$ also satisfies the conclusion (sentence), we'll say that the $KB$ **"entails" the conclusion (sentence)**.
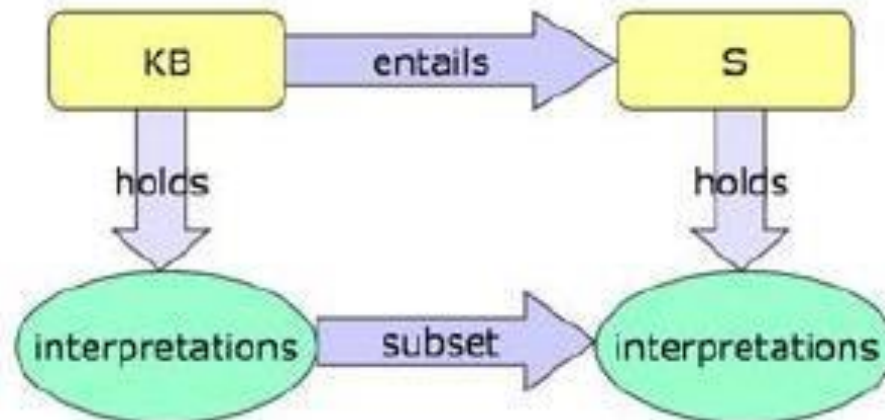
$$\alpha \models \beta \text{ if and only if } M(\alpha) \subseteq M(\beta)$$

# Entailment

- We have to enumerate all interpretations
- Those interpretations are selected where all elements of $KB$ are true.
- Check if the sentence $S$ is true for all those interpretations.



**Too many intpretations in general!**

# Entailment and Proof

▸ Proof is a way to test whether a $KB$ entails a sentence without enumerating too many interpretations.

▸ Proof is basically a sequence of sentences.

▸ Proof is a chain of conclusions that leads to the desired goal.
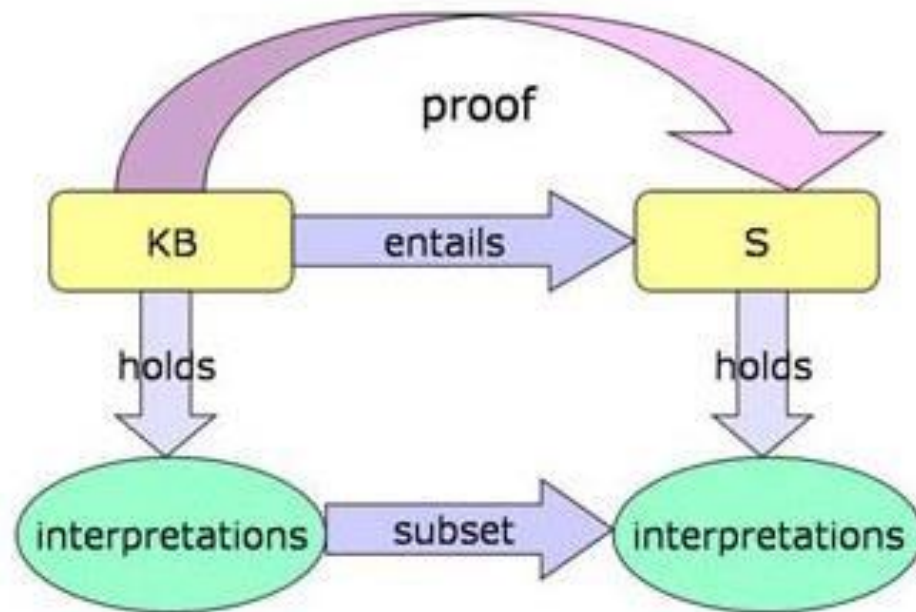
# Inference

- *If an inference algorithm $i$ can derive $\alpha$ from $KB$, we write*

$$KB \vdash_i \alpha$$

- which is pronounced "$\alpha$ **is derived from $KB$ by $i$**" or "$i$ **derives $\alpha$ from $KB$**."

# Inference Rules and Proofs

Some of Inference rules can be can be described as,

**Modus Ponens**

▸ If the sentences $\alpha \Rightarrow \beta$ and $\alpha$ are known to be true, then ***modus ponens*** lets us infer $\beta$.

$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

**Modus Tollens**

▸ Under the inference rule ***modus tollens***, if $\alpha \Rightarrow \beta$ is known to be true and $\beta$ is known to be false, we can infer $\neg\alpha$.

$$\frac{\alpha \rightarrow \beta \quad \neg \beta}{\neg \alpha}$$

# Inference Rules and Proofs

**And Elimination:**

▸ And elimination allows us to infer the truth of either of the conjuncts from the truth of a conjunctive sentence. For instance, α ∧ β lets us conclude α and β are true.

**And Introduction:**

$$\frac{\alpha \wedge \beta}{\alpha}$$

▸ And introduction lets us infer the truth of a conjunction from the truth of its conjuncts. For instance, if α and β are true, then α ∧ β is true.

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta}$$

# Inference Rules and Proofs

▸ Inference rules can be applied to derive a proof.

$$\frac{\begin{array}{c} \alpha \to \beta \\ \alpha \end{array}}{\beta}$$

Modus ponens

$$\frac{\begin{array}{c} \alpha \to \beta \\ \neg \beta \end{array}}{\neg \alpha}$$

Modus tolens

$$\frac{\begin{array}{c} \alpha \\ \beta \end{array}}{\alpha \wedge \beta}$$

And-introduction

$$\frac{\alpha \wedge \beta}{\alpha}$$

And-elimination

# Example

▸ Consider there are 3 sentences in our knowledge base, $P \wedge Q$, $P \rightarrow R$ and $(Q \wedge R) \rightarrow S$. **We have to prove S**.

| Step | Formula | Derivations |
|------|---------|-------------|
| 1 | $P \wedge Q$ | **Given** |
| 2 | $P \rightarrow R$ | **Given** |
| 3 | $(Q \wedge R) \rightarrow S$ | **Given** |
| 4 | **P** | **1 And-Elimination** |
| 5 | **R** | **2,4 Modes Ponens** |
| 6 | **Q** | **1 And-Elimination** |
| 7 | $Q \wedge R$ | **6,5 And-Introduction** |
| 8 | **S** | **3,7 Modes Ponens** |

# Example ... 2

▸ Consider there are 5 sentences in our knowledge base as given below.

▸ **We have to prove R**.

| Step | Formula | Derivations |
|------|---------|-------------|
| 1 | P → Z | **Given** |
| 2 | (P ∧ R) → S | **Given** |
| 3 | (Q ∧ Z) → R | **Given** |
| 4 | Z → Q | **Given** |
| 5 | P ∧ S | **Given** |
| **6** | **P** | **5 And-Elimination** |
| **7** | **Z** | **6,1 Modes Ponens** |
| **8** | **Q** | **4,7 Modes Ponens** |
| **9** | Q ∧ Z | **7,8 And introduction** |
| **10** | **R** | **3,9 Modes Ponens** |

# Propositional Resolution

# Propositional Resolution

- Resolution is one method for **automated theorem proving**.

- It helps logical agents **to reason** about the world.

- It helps logical agents **to prove new theorems**, and therefore helps them **to add to their knowledge**.

- Resolution requires all sentences to be first written in a special form - **conjunctive normal form**.

# Propositional Resolution

▸ Input a knowledge base and an expression.

▸ It **negates the expression**, adds that to the knowledge base, and finds a **contradiction** if one exists

▸ If it **finds** a contradiction, then the negated statement is **false**.

▸ Therefore, the original statement **must be true.**

# Conjunctive Normal Form

▸ A series of "conjunctions" (clauses joined together by "∧").

$$(A \lor B \lor \lnot C) \land (B \lor D) \land (\lnot A) \land (B \lor C)$$

**clauses**

▸ A, B,  ¬C are **literals**.

▸ Inside the brackets, we only have "∨(OR)",   "¬ (NOT)" symbols.

▸ *There must be no "implies" (e) symbols anywhere.*

# Conjunctive Normal Form

‣ **<u>Resolution algorithm</u>** 'resolves' clauses

‣ Each pair of clauses that contains **<u>complementary literals</u>** is resolved. Complementary literals have the property that one negates the other

  ◦ P, ¬P

  ◦ the clause $B_{1,1} \lor \neg B_{1,1} \lor P_{1,2}$ is equivalent to $True \lor P_{1,2}$ which is equivalent to $True$. Deducing that $True$ is true is not helpful.

‣ Therefore, *any clause in which two complementary literals appear can be discarded*.

‣ Each clause is a requirement that must be satisfied.

‣ It can be satisfied in multiple ways

# Converting into CNF

# Converting into CNF

▸ The first step is **to eliminate single and double arrows** using their definitions.

$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{implication elimination}$$

▸ The next step is to **drive in negation**, De Morgan Laws

$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{De Morgan}$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{De Morgan}$$

▸ The third step is to **distribute OR over AND**;

$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge$$

# Converting into CNF ... Example

$$(A \vee B) \rightarrow (C \rightarrow D)$$

1. Eliminate arrows

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

2. Drive in negations

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

3. Distribute

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

# Resolution

▸ Resolution Rule:

Resolution rule:

$$\frac{\alpha \vee \beta}{\neg \beta \vee \gamma}{\alpha \vee \gamma}$$

▸ Wumpus-world Example

$$\frac{P_{1,1} \vee P_{3,1}, \\ \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

# Resolution

Propositional Resolution:

▸ Convert all sentances into CNF

▸ Negate the desired conclusion (converted to CNF)

▸ Apply resolution rule until either

1. Derive false (contradiction)

   ◦ The conclusion follows from the axioms

2. Can not apply anymore,

   ◦ The conclusion can not be proved from axioms

**Propositional Resolution is sound and complete.**

# Resolution Example

▸ Consider we are given "P or Q", "P implies R" and "Q implies R". We would like to conclude R from these three axioms.

| 1 | P ∨ Q |
|---|-------|
| **2** | P → R |
| **3** | Q → R |

| Steps | Formula | Derivations |
|-------|---------|-------------|
| **1** | P ∨ Q | **Given** |
| **2** | ¬P ∨ R | **Given** |
| **3** | ¬Q ∨ R | **Given** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Resolution Example

▸ Consider we are given "P or Q", "P implies R" and "Q implies R". We would like to conclude R from these three axioms

| 1 | P ∨ Q |
|---|---|
| 2 | P → R |
| 3 | Q → R |

false v R

¬ R v false

──────────

false v false

| Steps | Formula | Derivations |
|---|---|---|
| 1 | P ∨ Q | **Given** |
| 2 | ¬P ∨ R | **Given** |
| 3 | ¬Q ∨ R | **Given** |
| **4** | **¬R** | **Negated Conclusion** |
| **5** | Q ∨ R | **1,2 Resolution Rule** |
| **6** | **¬P** | **2,4** |
| **7** | **¬Q** | **3,4** |
| **8** | **R** | **5,7** |
| **9** | **.** | **4,8** |

# Resolution Example

▸ Consider we are given "P or Q", "P implies R" and "Q implies R". We would like to conclude R from these three axioms

| 1 | P∨Q |
|---|-----|
| 2 | P→R |
| 3 | Q→R |

false v R
¬ R v false
―――――――――
false v false

| Steps | Formula | Derivations |
|-------|---------|-------------|
| 1 | P∨Q | **Given** |
| 2 | ¬P∨R | **Given** |
| 3 | ¬Q∨R | **Given** |
| 4 | **¬R** | **Negated Conclusion** |
| 5 | Q∨R | **1,2 Resolution Rule** |
| 6 | **¬P** | **2,4** |
| 7 | **¬Q** | **3,4** |
| 8 | **R** | **5,7** |
| 9 | **.** | **4,8** |

# Resolution Example 2

▶ Consider we are given following table. We would like to conclude R from these three axioms

| 1 | $(P \rightarrow Q) \rightarrow Q$ |
|---|---|
| 2 | $(P \rightarrow P) \rightarrow R$ |
| 3 | $(R \rightarrow S) \rightarrow \neg(S \rightarrow Q)$ |

# Resolution Example 2

▸ Consider we are given following table. We would like to conclude R from these three axioms

| 1 | (P → Q) → Q |
|---|---|
| 2 | (P → P) →R |
| 3 | (R → S ) → ¬(S → Q) |

| Step | Formula &Derivations |
|---|---|
| 1 | (P e  Q) e  Q |
| 2 | (¬P ∨ Q) e  Q |
| 3 | ¬(¬P ∨ Q)  ∨  Q |
| 4 | (P ∧ ¬Q) ∨ Q |
| 5 | (P ∨ Q) ∧ (¬ Q ∨ Q) |
| 6 | (P ∨ Q) |

# Resolution Example 2

▸ Consider we are given following table. We would like to conclude R from these three axioms

| 1 | (P → Q) → Q |
|---|---|
| 2 | (P → P) →R |
| 3 | (R → S ) → ¬(S → Q) |

| Step | Formula &Derivations |
|------|----------------------|
| 1 | (P e  P) e  R |
| 2 | (¬P ∨ P) e  R |
| 3 | ¬(¬P ∨ P) ∨ R |
| 4 | (P ∧ ¬P) ∨ R |
| 5 | (P ∨ R) ∧ (¬ P ∨ R) |

# Resolution Example 2

▸ Consider we are given following table. We would like to conclude R from these three axioms

| 1 | (P → Q) → Q |
|---|---|
| 2 | (P → P) → R |
| 3 | (R → S) → ¬(S → Q) |

| Step | Formula & Derivations |
|------|------------------------|
| 1 | (R e  S ) e ¬(S e  Q**)** |
| 2 | **(**¬R ∨ S**)** e  ¬(¬S ∨ Q) |
| 3 | **¬(**¬R ∨ S**)** ∨ ¬(¬S ∨ Q) |
| 4 | (R ∧ ¬S) ∨ (S ∧ ¬Q) |
| 5 | (R ∨ S) ∧ (R ∨ ¬Q) ∧ (¬S ∨ S ) ∧ (¬S ∨ **¬Q)** |
| 6 | (R ∨ S) ∧ (R ∨ ¬Q) ∧ (¬S ∨ **¬Q)** |

# Resolution Example 2

| Step | Formula | Derivations |
|---|---|---|
| 1 | (P ∨ **Q)** | |
| 2 | (P ∨ **R)** | |
| 3 | (¬P ∨ R) | |
| 4 | (R ∨ S) | |
| 5 | (R ∨ ¬Q) | |
| 6 | **(¬S ∨ ¬Q)** | |
| 7 | **¬R** | **Negation** |
| | | |
| | | |
| | | |
| | | |
| | | |

| | |
|---|---|
| 1 | (P → Q) → Q |
| 2 | (P → P) →R |
| 3 | (R → S ) → ¬(S → Q**)** |

# Resolution Example 2

| 1 | $(P \rightarrow Q) \rightarrow Q$ |
|---|---|
| 2 | $(P \rightarrow P) \rightarrow R$ |
| 3 | $(R \rightarrow S) \rightarrow \neg(S \rightarrow Q)$ |

| Step | Formula | Derivations |
|------|---------|-------------|
| 1 | $(P \vee Q)$ | |
| 2 | $(P \vee R)$ | |
| 3 | $(\neg P \vee R)$ | |
| 4 | $(R \vee S)$ | |
| 5 | $(R \vee \neg Q)$ | |
| 6 | $(\neg S \vee \neg Q)$ | |
| 7 | $\neg R$ | Negation |
| 8 | S | 4,7 |
| 9 | $\neg Q$ | 6,8 |
| 10 | P | 1,9 |
| 11 | R | 3,10 |
| 12 | . | 7,11 |

# Proof Strategies

▸ **Unit Preference:** Prefer a resolution step involving a unit clause  (*clause with one literal*)

  ▸ Produce a shorter clause which is good since we are trying to produce a zero-length clause, that is, a contradiction.

▸ **Set of support:** Choose *a resolution involving the negated goal* or any clause derived from the negated goal

  ◦ We're trying to produce a contradiction that follows from the negated goal, so these are "relevant" clauses.

  ◦ If a contradiction exists, one can find it using the set-of-support strategy.

# Reading Material

▸ **Artificial Intelligence,** A Modern Approach

**Stuart J. Russell and Peter Norvig**

◦ **Chapter 7.**