CS 1571 Introduction to AI
Lecture 11

Propositional logic (cont).
First order logic.

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Administration

- **PS-4:**
  - **Due on Thursday, October 2, 2003**
  - **Resolution-refutation** covered today

Logical inference problem

Logical inference problem:

- **Given:**
  - a knowledge base KB (a set of sentences) and
  - a sentence α (called **a theorem)**,
- **Does a KB semantically entail** α ? $KB \models \alpha$

In other words: In all interpretations in which sentences in the KB are true, is also α true?

Three approaches:

- **Truth-table approach**
- **Inference rules**
- **Conversion to the inverse SAT problem**
  - **Resolution-refutation**

Truth-table approach

Problem: $KB \models \alpha$?

- We need to check all possible interpretations for which the KB is true (models of KB) whether α is true for each of them

Truth tables:

- enumerate truth values of sentences for all possible interpretations (assignments of True/False to propositional symbols) and check

Example:

		KB		α	
P	Q	$P \vee Q$	$P \Leftrightarrow Q$	$(P \vee \neg Q) \wedge Q$	
True	*True*	*True*	*True*	*True*	✓
True	*False*	*True*	*False*	*False*	
False	*True*	*True*	*False*	*False*	
False	*False*	*False*	*True*	*False*	

Inference rules approach.

Motivation: **we do not want to blindly generate and check all interpretations !!!**

Inference rules:

- Represent **sound inference patterns** repeated in inferences
- Application of many inference rules allows us to infer new sound conclusions and hence prove theorems

- **An example of an inference rule: Modus ponens**

$$\frac{A \Rightarrow B, \quad A}{B}$$

← **premise**

← **conclusion**

Example. Inference rules approach.

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1.	$P \wedge Q$	
2.	$P \Rightarrow R$	
3.	$(Q \wedge R) \Rightarrow S$	
4.	P	**From 1 and And-elim**
5.	R	**From 2,4 and Modus ponens**
6.	Q	**From 1 and And-elim**
7.	$(Q \wedge R)$	**From 5,6 and And-introduction**
8.	S	**From 7,3 and Modus ponens**

Proved: S

Example. Inference rules approach.

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

Nondeterministic steps

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P **From 1 and And-elim**
5. R **From 2,4 and Modus ponens**
6. Q **From 1 and And-elim**
7. $(Q \wedge R)$ **From 5,6 and And-introduction**
8. S **From 7,3 and Modus ponens**

Proved: S

Logic inferences and search

Inference rule method as a search problem:

- **State:** a set of sentences that are known to be true
- **Initial state**: a set of sentences in the KB
- **Operators**: applications of inference rules
  - Allow us to add new sound sentences to old ones
- **Goal state:** a theorem α is derived from KB

Logic inference:

- **Proof:** A sequence of sentences that are immediate consequences of applied inference rules
- **Theorem proving:** process of finding a proof of theorem

Satisfiability (SAT) problem

Determine whether a sentence in the conjunctive normal form (CNF) is satisfiable (i.e. can evaluate to true)

$$(P \vee Q \vee \neg R) \wedge (\neg P \vee \neg R \vee S) \wedge (\neg P \vee Q \vee \neg T) \ldots$$

It is an instance of a constraint satisfaction problem:

- **Variables:**
  - Propositional symbols (*P, R, T, S*)
  - Values: ***True, False***
- **Constraints:**
  - Every conjunct must evaluate to true, at least one of the literals must evaluate to true
- **Why is this important?** All techniques developed for CSPs can be applied to solve the logical inference problem !!

Logic inference problem and satisfiability

Inference problem:

- we want to show that a sentence α is entailed by KB

Satisfiability:

- The sentence is satisfiable if there is some assignment (interpretation) under which the sentence evaluates to true

Connection:

$KB \models \alpha$ if and only if

$(KB \wedge \neg \alpha)$ is **unsatisfiable**

Consequences:

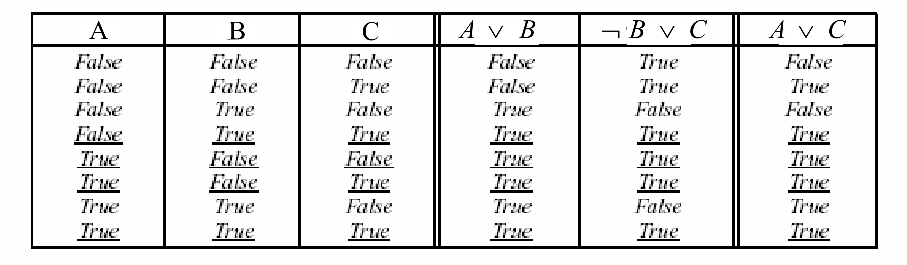
- programs for solving **SAT problems** can be used to solve the inference problem
- **SAT problem:** logical formulae in CNF

Resolution rule

Resolution rule

- A sound inference rule that **'fits'** the **CNF**

$$\frac{A \vee B, \quad \neg A \vee C}{B \vee C}$$



A	B	C	$A \vee B$	$\neg B \vee C$	$A \vee C$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

Resolution rule

Resolution rule

- sound inference rule that **'fits'** the **CNF**

$$\frac{A \vee B, \quad \neg A \vee C}{B \vee C}$$

- It is **complete (refutation complete) for the logical inference problem in propositional logic**
  - Repeated application of the resolution rule to a KB in CNF can be used to answer the logical inference problem $KB \models \alpha$
  - Uses refutation proofs to assure the completeness

Resolution.

Why refutation?

- Repeated application of the resolution rule to a KB in CNF may fail to derive new valid sentences

  Example:

  We know: $(A \wedge B)$ We want to show: $(A \vee B)$

  Resolution rule fails to derive it (**incomplete ??)**

Proof by contradiction :

  - **Disproving:** $KB, \neg \alpha$
  - **Proves the entailment** $KB \models \alpha$

- **Avoids the problem. How?**

  $(A \wedge B) \models (A \vee B)$

Resolution algorithm

Algorithm:

1. **Convert KB to the CNF form;**
2. **Apply iteratively the resolution rule** starting from
   $KB, \neg \alpha$ (in the CNF form)
3. **Stop when**:
   - Contradiction (empty clause) is reached:
     - $A, \neg A \rightarrow \emptyset$
     - **proves the entailment.**
   - No more new sentences can be derived
     - **Rejects (disproves) the entailment**.

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

Step 1. convert KB to CNF:

- $P \wedge Q \longrightarrow P \wedge Q$
- $P \Rightarrow R \longrightarrow (\neg P \vee R)$
- $(Q \wedge R) \Rightarrow S \longrightarrow (\neg Q \vee \neg R \vee S)$

KB: $P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S)$

Step 2. Negate the theorem to prove it via refutation

$S \longrightarrow \neg S$

Step 3. Run resolution on the set of clauses

$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

Example. Resolution.

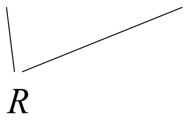
KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

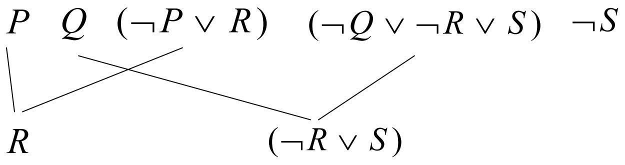
$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$



R

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

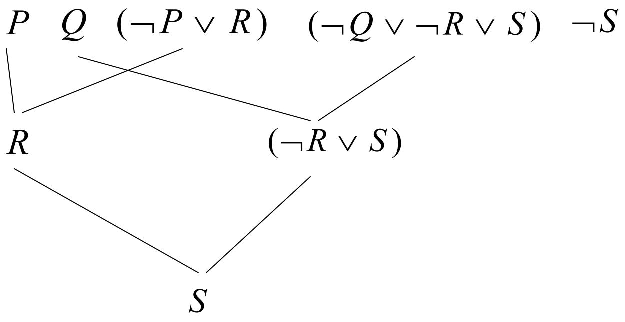


$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

$R \qquad (\neg R \vee S)$

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S



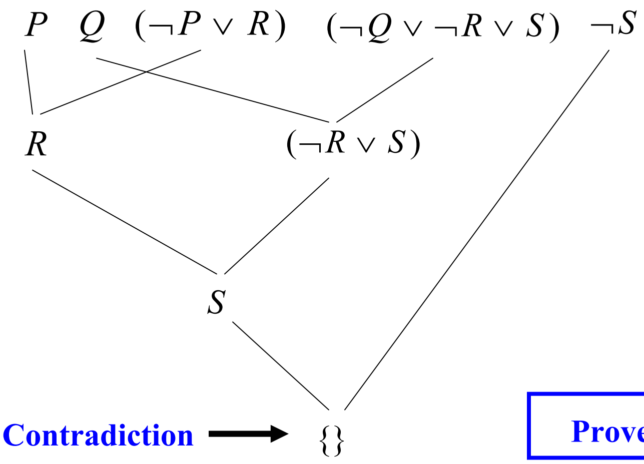
$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

$R \qquad (\neg R \vee S)$

S

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S



P Q $(\neg P \vee R)$ $(\neg Q \vee \neg R \vee S)$ $\neg S$

R $(\neg R \vee S)$

S

Contradiction → $\{\}$

Proved: S

KB in restricted forms

- If the sentences in the KB are restricted to some special forms other sound inference rules may become complete

Example:

- **Horn form (Horn normal form)**

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

Can be written also as: $(B \Rightarrow A) \wedge ((A \wedge C) \Rightarrow D)$

- **Modus ponens:**
  - is the “universal “(complete) rule for the sentences in the Horn form

$$\frac{A \Rightarrow B, \quad A}{B} \qquad \frac{A_1 \wedge A_2 \wedge \ldots \wedge A_k \Rightarrow B,\ A_1, A_2, \ldots A_k}{B}$$

KB in Horn form

- **Horn form:** a clause with at most one positive literal

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

- **Not all sentences in propositional logic can be converted into the Horn form**
- **KB in Horn normal form**:
  - Two types of propositional statements:
    - Implications: called **rules** $(B \Rightarrow A)$
    - Propositional symbols: **facts** B
- **Application of the modus ponens:**
  - Infers new facts from previous facts

$$\frac{A \Rightarrow B, \quad A}{B}$$

Forward and backward chaining

Two inference procedures based on **modus ponens** for **Horn KBs**:

- **Forward chaining**

  Idea: Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

- **Backward chaining (goal reduction)**

  Idea: To prove the fact that appears in the conclusion of a rule prove the premises of the rule. Continue recursively.

Both procedures are **complete for KBs in the Horn form** !!!

Forward chaining example

- **Forward chaining**

  Idea: Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

Assume the KB with the following rules and facts:

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

Theorem: E ?

Forward chaining example

Theorem: E

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

Forward chaining example

Theorem: E

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

Rule R1 is satisfied.

F4: C

Forward chaining example

Theorem: E

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

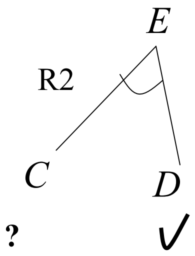
Rule R1 is satisfied.

F4: C

Rule R2 is satisfied.

F5: E ✓

Backward chaining example



KB: R1: $A \wedge B \Rightarrow C$

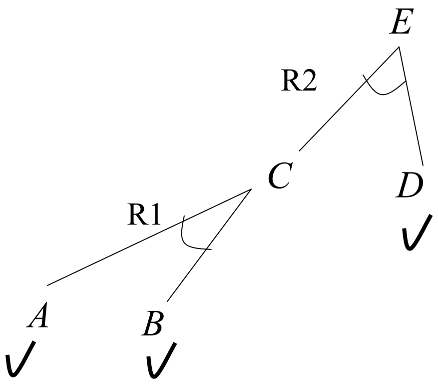
R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

- Backward chaining is more focused:
  - tries to prove the theorem only

Backward chaining example



KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

- Backward chaining is more focused:
  - tries to prove the theorem only

KB agents based on propositional logic

- Propositional logic allows us to build **knowledge-based agents** capable of answering queries about the world by infering new facts from the known ones
- **Example:** an agent for diagnosis of a bacterial disease

Facts: The stain of the organism is gram-positive
The growth conformation of the organism is chains

Rules: **(If)** The stain of the organism is gram-positive $\wedge$
The morphology of the organism is coccus $\wedge$
The growth conformation of the organism is chains
(Then) $\Rightarrow$ The identity of the organism is streptococcus

---

Limitations of propositional logic

World we want to represent and reason about consists of a number of objects with variety of properties and relations among them

Propositional logic:

- Represents statements about the world without reflecting this structure and without modeling these entities explicitly

Consequence:

- some knowledge is hard or impossible to encode in the propositional logic.
- Two cases that are hard to represent:
  - **Statements about similar objects, relations**
  - **Statements referring to groups of objects**.

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

Assume we have: *John is older than Mary*
Mary is older than Paul

To derive *John is older than Paul* we need:

John is older than Mary ∧ *Mary is older than Paul*
⇒ *John is older than Paul*

Assume we add another fact: *Jane is older than Mary*

To derive *Jane is older than Paul* we need:

Jane is older than Mary ∧ *Mary is older than Paul*
⇒ *Jane is older than Paul*

What is the problem?

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

Assume we have: *John is older than Mary*
Mary is older than Paul

To derive *John is older than Paul* we need:

John is older than Mary ∧ *Mary is older than Paul*
⇒ *John is older than Paul*

Assume we add another fact: *Jane is older than Mary*

To derive *Jane is older than Paul* we need:

Jane is older than Mary ∧ *Mary is older than Paul*
⇒ *Jane is older than Paul*

Problem: KB grows large

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

  For inferences we need:

  John is older than Mary ∧ *Mary is older than Paul*
  ⇒ *John is older than Paul*
  Jane is older than Mary ∧ *Mary is older than Paul*
  ⇒ *Jane is older than Paul*

- **Problem:** if we have many people and facts about their seniority we need represent many rules like this to allow inferences
- **Possible solution: ??**

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

  For inferences we need:

  John is older than Mary ∧ *Mary is older than Paul*
  ⇒ *John is older than Paul*
  Jane is older than Mary ∧ *Mary is older than Paul*
  ⇒ *Jane is older than Paul*

- **Problem:** if we have many people and facts about their seniority we need represent many rules like this to allow inferences
- **Possible solution: introduce variables**

  PersA *is older than* ***PersB*** ∧ ***PersB*** *is older than* ***PersC***
  ⇒ ***PersA*** *is older than* ***PersC***

Limitations of propositional logic

- **Statements referring to groups of objects require exhaustive enumeration of objects**
- **Example:**

Assume we want to express *Every student likes vacation*

Doing this in propositional logic would require to include statements about every student

John likes vacation $\wedge$
Mary likes vacation $\wedge$
Ann likes vacation $\wedge$
…

- **Solution:** Allow quantification in statements

First-order logic (FOL)

- More expressive than **propositional logic**

- **Eliminates deficiencies of PL by:**
  - Representing objects, their properties, relations and statements about them;
  - Introducing variables that refer to an arbitrary objects and can be substituted by a specific object
  - Introducing quantifiers allowing quantification statements over objects without the need to represent each of them separately

- **Predicate logic:** first-order logic without the quantification fix