CS 1571 Introduction to AI
Lecture 12

Propositional logic

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

Announcements

- **Homework assignment 4 due today**
- **Homework assignment 5 is out**
  - Programming and experiments
  - Tic-tac-toe player
  - Competition

Course web page:

http://www.cs.pitt.edu/~milos/courses/cs1571/

Knowledge representation

- **Knowledge representation**
  - **Objective:** express the knowledge about the world in a computer-tractable form
  - **Knowledge representation languages (KRLs)**

- **Inference procedures:**
  - A set of procedures that use the knowledge representational language (KRL) to infer new facts from known ones or answer a variety of KB queries. Typically require a search.

- **Last and this lecture:**
  - **use of Propositional logic as KRL**

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?

Sound and complete inference.

Inference is a process by which conclusions are reached.

- We want to implement the inference process on a computer !!

Assume an **inference procedure *i*** that

- derives a sentence α from the KB : $KB \vdash_i \alpha$

Properties of the inference procedure in terms of entailment

- **Soundness:** An inference procedure is **sound**

  If $KB \vdash_i \alpha$ then it is true that $KB \models \alpha$

- **Completeness:** An inference procedure is **complete**

  If $KB \models \alpha$ then it is true that $KB \vdash_i \alpha$

Solving logical inference problem

In the following:

How to design the procedure that answers:

$$KB \models \alpha \ ?$$

Three approaches:

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

Truth-table approach

A two steps procedure:

1. **Generate table for all possible interpretations**
2. Check whether the sentence α evaluates to true whenever KB evaluates to true

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	*True*	*True*				
True	*True*	*False*				
True	*False*	*True*				
True	*False*	*False*				
False	*True*	*True*				
False	*True*	*False*				
False	*False*	*True*				
False	*False*	*False*				

Truth-table approach

A two steps procedure:

1. Generate table for all possible interpretations
2. **Check whether the sentence α evaluates to true whenever *KB* evaluates to true**

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	*True*	*True*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*True*	*True*	*True*
True	*False*	*True*	*True*	*False*	*False*	*True*
True	*False*	*False*	*True*	*True*	*True*	*True*
False	*True*	*True*	*True*	*True*	*True*	*True*
False	*True*	*False*	*False*	*True*	*False*	*True*
False	*False*	*True*	*True*	*False*	*False*	*False*
False	*False*	*False*	*False*	*True*	*False*	*False*

Truth-table approach

A two steps procedure:

1. Generate table for all possible interpretations
2. **Check whether the sentence α evaluates to true whenever *KB* evaluates to true**

Example: $KB = (A \vee C) \wedge (B \vee \neg C) \qquad \alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α	
True	*True*	*True*	*True*	*True*	*True*	*True*	✓
True	*True*	*False*	*True*	*True*	*True*	*True*	✓
True	*False*	*True*	*True*	*False*	*False*	*True*	
True	*False*	*False*	*True*	*True*	*True*	*True*	✓
False	*True*	*True*	*True*	*True*	*True*	*True*	✓
False	*True*	*False*	*False*	*True*	*False*	*True*	
False	*False*	*True*	*True*	*False*	*False*	*False*	
False	*False*	*False*	*False*	*True*	*False*	*False*	

Truth-table approach

$KB = (A \vee C) \wedge (B \vee \neg C) \qquad \alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	*True*	*True*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*True*	*True*	*True*
True	*False*	*True*	*True*	*False*	*False*	*True*
True	*False*	*False*	*True*	*True*	*True*	*True*
False	*True*	*True*	*True*	*True*	*True*	*True*
False	*True*	*False*	*False*	*True*	*False*	*True*
False	*False*	*True*	*True*	*False*	*False*	*False*
False	*False*	*False*	*False*	*True*	*False*	*False*

KB entails α

- The **truth-table approach** is **sound and complete** for the propositional logic!!

Limitations of the truth table approach

$$KB \models \alpha \;?$$

- **What is the computational complexity of the truth table approach?**

Exponential in the number of the propositional symbols

2^n Rows in the table has to be filled

– the truth table is **exponential** in the number of propositional symbols (we checked all assignments)

Limitation of the truth table approach

$$KB \models \alpha \;?$$

Problem with the truth table approach:

- the truth table is **exponential** in the number of propositional symbols (we checked all assignments)

How to make the process more efficient?

Observation: KB is true only on a small subset interpretations

Solution: inference rules approach

- **start from entries for which KB is *True*.**
- **generate new (sound) sentences from the existing ones**

Inference rules approach

Approach:

- start from KB
- infer new sentences that are true from existing KB sentences
- Repeat till alpha is proved (inferred true) or no more sentences can be proved

Rules:

– let us generate new (sound) sentences from the existing ones

- **Equivalence rules:**
  - Known logical equivalences
- **Inference rules:**
  - Represent sound "local" inference patterns repeated in inferences

---

Logical equivalence

- **Definition**: The propositions P and Q are called **logically equivalent** if P ↔ Q is a tautology (alternately, if they have the same truth table). The notation **P <=> Q** denotes P and Q are logically equivalent.

A	B	**A → B**	**¬A → ¬B**	**(A → B)** <-> **(¬A → ¬B)**
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

Important logical equivalences

- **Identity**
  - $p \wedge T \iff p$
  - $p \vee F \iff p$
- **Domination**
  - $p \vee T \iff T$
  - $p \wedge F \iff F$
- **Idempotent**
  - $p \vee p \iff p$
  - $p \wedge p \iff p$

Important logical equivalences

- **Double negation**
  - $\neg(\neg p) \iff p$
- **Commutative**
  - $p \vee q \iff q \vee p$
  - $p \wedge q \iff q \wedge p$
- **Associative**
  - $(p \vee q) \vee r \iff p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \iff p \wedge (q \wedge r)$

Important logical equivalences

- **Distributive**
  - $p \lor (q \land r) \iff (p \lor q) \land (p \lor r)$
  - $p \land (q \lor r) \iff (p \land q) \lor (p \land r)$

- **De Morgan**
  - $\neg(p \lor q) \iff \neg p \land \neg q$
  - $(p \land q) \iff \neg p \lor \neg q$

- **Other useful equivalences**
  - $p \lor \neg p \iff T$
  - $p \land \neg p \iff F$
  - **$p \rightarrow q \iff (\neg p \lor q)$**

Inference rules

- **Modus ponens**

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

← **premise**

← **conclusion**

- If both sentences in the premise are true then conclusion is true.
- The modus ponens inference rule is **sound.**
  - We can prove this through the truth table.

A	B	$A \Rightarrow B$
False	*False*	*True*
False	*True*	*True*
True	*False*	*False*
True	$\boxed{\textit{True}}$	*True*

Inference rules for logic

- **And-elimination**

$$\frac{A_1 \land A_2 \land \quad A_n}{A_i}$$

- **And-introduction**

$$\frac{A_1, A_2, \quad A_n}{A_1 \land A_2 \land \quad A_n}$$

- **Or-introduction**

$$\frac{A_i}{A_1 \lor A_2 \lor \ldots A_i \lor \quad A_n}$$

Inference rules for logic

- **Unit resolution**

$$\frac{A \lor B, \quad \neg A}{B}$$

A special case of

- **Resolution**

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

- All of the above inference rules **are sound.** We can prove this through the truth table, similarly to the **modus ponens** case.

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. $(Q \wedge R) \Rightarrow S$

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**

$$\frac{A_1 \wedge A_2 \wedge \quad A_n}{A_i}$$

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
5. R

From 2,4 and Modus ponens

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

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
5. R
6. Q

From 1 and And-elim

$$\frac{A_1 \wedge A_2 \wedge \quad A_n}{A_i}$$

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
5. R
6. Q
7. $(Q \wedge R)$

From 5,6 and And-introduction

$$\frac{A_1, A_2, \ldots A_n}{A_1 \wedge A_2 \wedge \ldots A_n}$$

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
5. R
6. Q
7. $(Q \wedge R)$
8. S

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

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

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**

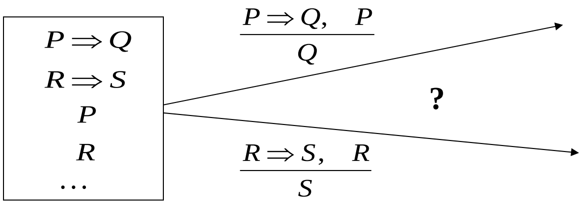
Proved: S

Logic inferences and search

- To show that theorem α holds for a KB
  - we may need to apply a number of sound inference rules

Problem: many possible rules to can be applied next

Looks familiar?



$P \Rightarrow Q$

$R \Rightarrow S$

P

R

$\dots$

$$\frac{P \Rightarrow Q, \quad P}{Q}$$

?

$$\frac{R \Rightarrow S, \quad R}{S}$$

This is an instance of a search problem:

Truth table method (from the search perspective):

- blind enumeration and checking

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

Normal forms

Problems:

- Too many different rules one can apply
- Many new sentence are just equivalent sentences

Question:

- Can we simplify inferences using one of the normal forms?

Normal forms

Conjunctive normal form (CNF)

- conjunction of clauses (clauses include disjunctions of literals)

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

Disjunctive normal form (DNF)

- Disjunction of terms (terms include conjunction of literals)

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

Conversion to a CNF

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

1. Eliminate $\Rightarrow, \Leftrightarrow$

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

2. Reduce the scope of signs through DeMorgan Laws and double negation

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

3. Convert to CNF using the associative and distributive laws

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

and

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

Inferences in CNF

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

1. Eliminate $\Rightarrow, \Leftrightarrow$

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

2. Reduce the scope of signs through DeMorgan Laws and double negation

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

3. Convert to CNF using the associative and distributive laws

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

and

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

Resolution rule

Resolution rule

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

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

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 for the KB expressed in the CNF form
- But unfortunately not complete
  - – 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 ??)**

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

---

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

- **A logical inference problem can be solved as a CSP problem. Why?**

Inference problem and satisfiability

Inference problem:

- we want to show that the 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:

- inference problem is NP-complete
- programs for solving the SAT problem can be used to solve the inference problem

---

Resolution rule

When applied directly to KB in CNF to infer α:

- **Incomplete:** repated 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 is incomplete

A trick to make things work:

- **proof by contradiction**
  - **Disproving:** $KB \wedge \neg \alpha$
  - **Proves the entailment** $KB \models \alpha$

  Resolution rule is refutation complete

Resolution algorithm

Algorithm:

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

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

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

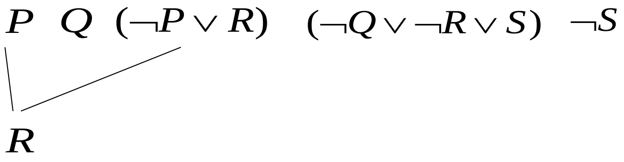
Example. Resolution.

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

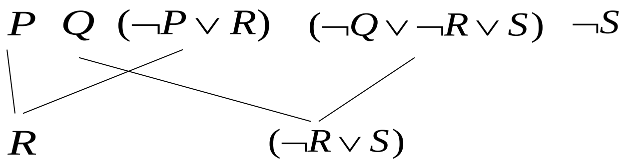


$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

R

Example. Resolution.

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

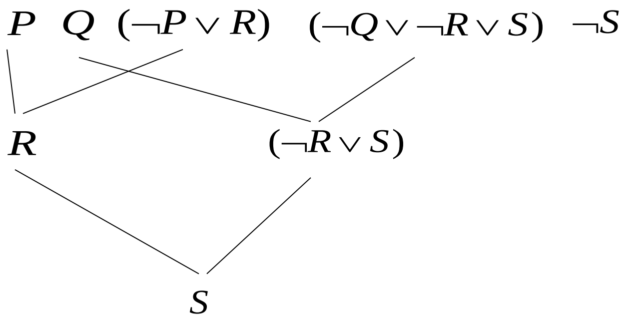


P Q $(\neg P \vee R)$ $(\neg Q \vee \neg R \vee S)$ $\neg S$

R $(\neg R \vee S)$

Example. Resolution.

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



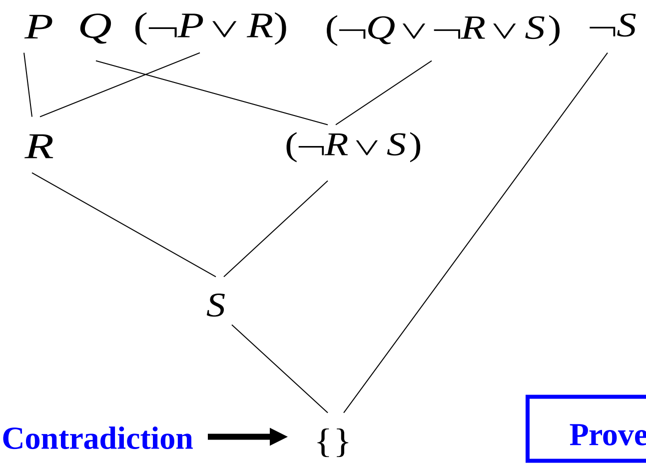
P Q $(\neg P \vee R)$ $(\neg Q \vee \neg R \vee S)$ $\neg S$

R $(\neg R \vee S)$

S

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

Properties of inference solutions

- **Truth-table approach**
  - **Blind**
  - **Exponential in the number of variables**
- **Inference rules**
  - **More efficient**
  - **Many inference rules to cover logic**
- **Conversion to SAT - Resolution refutation**
  - **More efficient**
  - **Sentences must be converted into CNF**
  - **One rule – the resolution rule - is sufficient to perform all inferences**