CS 2740 Knowledge Representation
Lecture 5

Propositional logic II.

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

Propositional logic. Syntax

Syntax:

- **Symbols (alphabet)** in P:
  - **Constants**: ***True, False***
  - **Propositional symbols**

    Examples:
    - P
    - *Pitt is located in the Oakland section of Pittsburgh.,*
    - *It rains outside,* etc.
  - **A set of connectives**:

$$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$$

Propositional logic. Syntax

Sentences in the propositional logic:

- **Atomic sentences:**
  - **Constructed from constants and propositional symbols**
  - True, False are (atomic) sentences
  - P, Q or *Light in the room is on, It rains outside* are (atomic) sentences
- **Composite sentences:**
  - **Constructed from valid sentences via connectives**
  - If A, B are sentences then
    $\neg A \quad (A \wedge B) \quad (A \vee B) \quad (A \Rightarrow B) \quad (A \Leftrightarrow B)$
    or $(A \vee B) \wedge (A \vee \neg B)$
    are sentences

Propositional logic. Semantics.

The semantic gives the meaning to sentences.

the semantics in the propositional logic is defined by:

1. **Interpretation of propositional symbols and constants**
   - Semantics of atomic sentences
2. **Through the meaning of connectives**
   - Meaning (semantics) of composite sentences

Semantic: propositional symbols

A **propositional symbol represents:**

- a statement about the world that is either true or false

  Examples:
  - *Pitt is located in the Oakland section of Pittsburgh*
  - *It rains outside*
  - *Light in the room is on*
- An **interpretation** maps propositional symbols to one of the two values: ***True (T),*** or ***False (F)***, depending on whether the symbol is satisfied in the world

  I: *Light in the room is on -> **True**, It rains outside -> **False***

  I': *Light in the room is on -> **False**, It rains outside -> **False***

Semantic: propositional symbols

The **meaning (value)** of the propositional symbol for a specific interpretation is given by its interpretation

I: *Light in the room is on -> **True**, It rains outside -> **False***

V(Light in the room is on, **I***) = **True***

V(It rains outside, **I***) = **False***

I': *Light in the room is on -> **False**, It rains outside -> **False***

V(Light in the room is on, **I'***) = **False***

Semantics: constants

- **The meaning (truth) of constants:**
  - True and False constants are always (under any interpretation) assigned the corresponding ***True,False*** value

$$\left.\begin{array}{l} V(True, \mathbf{I}) = \textbf{\textit{True}} \\ \\ V(False, \mathbf{I}) = \textbf{\textit{False}} \end{array}\right\} \quad \text{For any interpretation } \mathbf{I}$$

Semantics: composite sentences

- **The meaning (truth value) of complex propositional sentences.**
  - Determined using the standard rules of logic:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
True	***True***	***False***	***True***	***True***	***True***	***True***
True	***False***	***False***	***False***	***True***	***False***	***False***
False	***True***	***True***	***False***	***True***	***True***	***False***
False	***False***	***True***	***False***	***False***	***True***	***True***

Rows define all possible interpretations (worlds) I

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?

Inference procedures

Inference is a process by which conclusions are reached.

Our goal: We want to implement the inference process on a computer !!

Question: Is there a procedure (program) that can decide this problem in a finite number of steps?

Answer: **Yes**. Logical inference problem for the propositional logic is **decidable**.

Inference procedures: properties

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

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

Properties of the inference procedure wrt 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

How to design the procedure that answers:

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

Three approaches:

- **Truth-table approach**
- **Inference rules**

$\left.\right\}$ Covered last lecture

- **Conversion to the SAT problem** ⟸
  - **Resolution-refutation** ⟸

Inference rules for logic

- **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	$\boxed{True}$	*True*

Example. Inference rules approach.

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

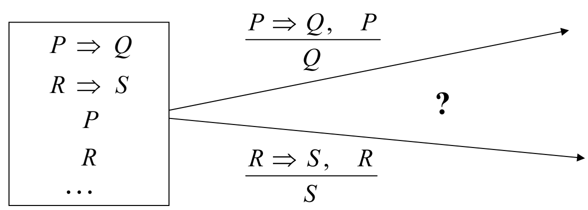
1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P — **From 1 and And-elim**
5. R — **From 2,4 and Modus ponens**
6. Q — **From 1 and And-elim**
7. $(Q \wedge R)$ — **From 5,6 and And-introduction**
8. S — **From 7,3 and Modus ponens**

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 can be applied in the next step



This is an instance of a search problem:

Truth table method (from the search perspective):

- blind enumeration and checking

Sentence transformations

Propositional logic:

- A sentence may include connectives: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- A sentence may consists of multiple nested sentences

Do we need all operators to represent a complex sentence?

No. We can rewrite a sentence in PL using an equivalent sentence with just $\neg, \wedge, \vee$ operators

Is it possible to limit the depth of the sentence structure?

Yes. Example: Normal forms.

Normal forms

Sentences in the propositional logic can be transformed into one of the **normal forms.** This can simplify the inferences.

Normal forms used:

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

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

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 (CSP):

- **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

Solving SAT

- An NP-complete problem
- **Methods:**
- **Backtracking:** equivalent to the depth first search
  - Pick the variable, then pick its value (T or F)
  - Continue till all variables are assigned or till the partial assignment makes the sentence False
- **Iterative optimization methods:**
  - Start from an arbitrary assignment of T,F values to symbols
  - Flip T,F values for one variable
  - **Heuristics:** prefer assignments that make more clauses true
  - Walk-SAT, simulated-annealing procedures

Inference problem and satisfiability

Logical inference problem:

- we want to show that the sentence α is entailed by KB
- For all interpretations for which KB is true is α also true?

Satisfiability (SAT) problem:

- Is there is some assignment (interpretation) under which the sentence evaluates to true?

Is it possible to formulate a logical inference problem as a satisfiability problem?

Inference problem and satisfiability

Logical inference problem:

- For all interpretations for which KB is true is α also true?

Satisfiability:

- Is there is some assignment (interpretation) under which a 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

Inferences with the CNF

- Can inference rule approach benefit from the CNF conversion?
- Is there an inference rule that is sufficient to support all inferences for the KB in the CNF form?

A rule that seems to fit very well the CNF form is the resolution rule.

Resolution rule

Resolution rule

- sound inference rule that works for the KB in the CNF form

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

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*

Inferences with the resolution rule

- **Resolution rule is sound**
- **Is it complete?**
  - Is it possible to use it such that we start from the KB, then prove the new facts and eventually prove the theorem if it is entailed?
- Not always = incomplete
- 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 inferences with the resolution rule

Conversion to SAT:

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

Important:

- The resolution rule is sufficient to determine satisfiability/unsatisfiability of

$$(KB \wedge \neg \alpha)$$

For any KB and theorem in the CNF

We say the resolution rule is **refutation complete**.

Resolution algorithm

Algorithm:

- **Convert KB to the CNF form;**
- **Apply iteratively the resolution rule** starting from $(KB \wedge \neg \alpha)$ (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

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

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

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

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

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

S

Contradiction → $\{\}$

Proved: S