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 \( \alpha \) (called a theorem),
• Does a KB semantically entail \( \alpha \)? \( KB \models \alpha \)
  In other words: In all interpretations in which sentences in
  the KB are true, is also \( \alpha \) 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 \( \alpha \) 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:

<table>
<thead>
<tr>
<th></th>
<th>( P )</th>
<th>( Q )</th>
<th>( P \lor Q )</th>
<th>( P \Leftrightarrow Q )</th>
<th>( (P \lor \neg Q) \land Q )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

CS 1571 Intro to AI
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**

\[ A \Rightarrow B, \quad A \quad \Rightarrow \quad \text{premise} \]
\[ B \quad \Rightarrow \quad \text{conclusion} \]

Example. Inference rules approach.

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

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

**Proved:** \( S \)
Example. Inference rules approach.

KB: \( P \land Q \quad P \Rightarrow R \quad (Q \land R) \Rightarrow S \)  \hspace{1cm} Theorem: \( S \)

1. \( P \land Q \)  \hspace{1cm} Non deterministic steps
2. \( P \Rightarrow R \)
3. \( (Q \land R) \Rightarrow S \)
4. \( P \)  \hspace{1cm} From 1 and And-elim
5. \( R \)  \hspace{1cm} From 2,4 and Modus ponens
6. \( Q \)  \hspace{1cm} From 1 and And-elim
7. \( (Q \land R) \)  \hspace{1cm} From 5,6 and And-introduction
8. \( S \)  \hspace{1cm} 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 \( \alpha \) 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)

\((P \lor Q \lor \neg R) \land (\neg P \lor \neg R \lor S) \land (\neg P \lor Q \lor \neg T)\) …

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 \(\alpha\) is entailed by \(KB\)

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

**Connection:**

\[ KB \models \alpha \quad \text{if and only if} \quad (KB \land \neg \alpha) \text{ is unsatisfiable} \]

**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

\[
\begin{align*}
A \lor B, & \quad \neg A \lor C \\
\hline
B \lor C
\end{align*}
\]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A \lor B</th>
<th>\neg B \lor C</th>
<th>A \lor C</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
Resolution.

Why refutation?
• Repeated application of the resolution rule to a KB in CNF may fail to derive new valid sentences

Example:
We know: \((A \land B)\) We want to show: \((A \lor B)\)
Resolution rule fails to derive it (incomplete ?)

Proof by contradiction:
– Disproving: \(KB, \neg \alpha\)
– Proves the entailment \(KB \models \alpha\)

• Avoids the problem. How?
\((A \land B) \models (A \lor B)\)

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:
  • \(A, \neg A \rightarrow \Box\)
  • proves the entailment.
– No more new sentences can be derived
  • Rejects (disproves) the entailment.
Example. Resolution.

KB: \((P \land Q) \land (P \Rightarrow R) \land [(Q \land R) \Rightarrow S]\)  Theorem: \(S\)

Step 1. convert KB to CNF:
- \(P \land Q \quad \rightarrow \quad P \land Q\)
- \(P \Rightarrow R \quad \rightarrow \quad (\neg P \lor R)\)
- \((Q \land R) \Rightarrow S \quad \rightarrow \quad (\neg Q \lor \neg R \lor S)\)

KB: \(P \quad Q \quad (\neg P \lor R) \quad (\neg Q \lor \neg R \lor S)\)

Step 2. Negate the theorem to prove it via refutation
\(S \quad \rightarrow \quad \neg S\)

Step 3. Run resolution on the set of clauses
\(P \quad Q \quad (\neg P \lor R) \quad (\neg Q \lor \neg R \lor S) \quad \neg S\)
Example. Resolution.

KB: \((P \land Q) \land (P \Rightarrow R) \land [(Q \land R) \Rightarrow S]\)   Theorem: \(S\)

\[
\begin{align*}
P & \hspace{0.5cm} Q & & \neg P \lor R & & \neg Q \lor \neg R \lor S & & \neg S \\
    \underline{R} & & & &
\end{align*}
\]
Example. Resolution.

KB: \((P \land Q) \land (P \Rightarrow R) \land [(Q \land R) \Rightarrow S]\)  Theorem: \(S\)

\[
P \quad Q \quad (\neg P \lor R) \quad (\neg Q \lor \neg R \lor S) \quad \neg S
\]

\[
R \quad (\neg R \lor S)
\]
Example. Resolution.

KB: \((P \land Q) \land (P \Rightarrow R) \land [(Q \land R) \Rightarrow S]\)

Theorem: \(S\)

\[
\begin{array}{ccc}
P & Q & (\neg P \lor R) & (\neg Q \lor \neg R \lor S) & \neg S \\
R & (\neg R \lor S) & S & \text{Contradiction} & \{\} & \text{Proved: } S
\end{array}
\]

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)**
  \[(A \lor \neg B) \land (\neg A \lor \neg C \lor D)\]
  Can be written also as: \((B \Rightarrow A) \land ((A \land C) \Rightarrow D)\)

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

\[
\frac{A \Rightarrow B, \ A}{B} \quad \frac{A_1 \land A_2 \land \ldots \land A_k \Rightarrow B, A_1, A_2, \ldots A_k}{B}
\]
KB in Horn form

- **Horn form**: a clause with at most one positive literal
  \[(A \lor \neg B) \land (\neg A \lor \neg C \lor D)\]
- 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
  \[
  \frac{A \Rightarrow B, \ A}{B}
  \]

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: \( A \land B \Rightarrow C \)
  - R2: \( C \land D \Rightarrow E \)
  - R3: \( C \land F \Rightarrow G \)

- **Facts:**
  - F1: \( A \)
  - F2: \( B \)
  - F3: \( D \)

**Theorem:** \( E \)?
Forward chaining example

Theorem: $E$

KB: R1: $A \land B \Rightarrow C$
    R2: $C \land D \Rightarrow E$
    R3: $C \land F \Rightarrow G$

F1: $A$
F2: $B$
F3: $D$

Rule R1 is satisfied.
F4: $C$

Rule R2 is satisfied.
F5: $E$
Backward chaining example

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 inferring 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 ∧
  The morphology of the organism is coccus ∧
  The growth conformation of the organism is chains
- **(Then)** ⇒ 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?

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:
  
  \[\text{John is older than Mary} \land \text{Mary is older than Paul}\]
  \[\Rightarrow \text{John is older than Paul}\]
  \[\text{Jane is older than Mary} \land \text{Mary is older than Paul}\]
  \[\Rightarrow \text{Jane is older than Paul}\]

• Problem: if we have many people and facts about their seniority
  we need represent many rules like this to allow inferences

• Possible solution: ??

\[\text{PersA is older than PersB} \land \text{PersB is older than PersC}\]
\[\Rightarrow \text{PersA is older than PersC}\]
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
  
  \[
  \begin{align*}
  John \text{ likes vacation} \land \\
  Mary \text{ likes vacation} \land \\
  Ann \text{ likes vacation} \land \\
  \vdots
  \end{align*}
  \]
  
- **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