Propositional logic

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?
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 \( \alpha \) from the KB: \( KB \vdash_i \alpha \)

Properties of the inference procedure in terms of entailment
• **Soundness:** An inference procedure is **sound**

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

If \( KB \vdash_i \alpha \) then it is true that \( KB \models \alpha \)

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 $\alpha$ evaluates to true whenever $KB$ evaluates to true

**Example:**

$KB = (A \lor C) \land (B \lor \neg C) \quad \alpha = (A \lor B)$

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$A \lor C$</th>
<th>$(B \lor \neg C)$</th>
<th>$KB$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</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>
Truth-table approach

A two steps procedure:
1. Generate table for all possible interpretations
2. Check whether the sentence $\alpha$ evaluates to true whenever $KB$ evaluates to true

Example: $KB = (A \lor C) \land (B \lor \neg C)$, $\alpha = (A \lor B)$

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$A \lor C$</th>
<th>$(B \lor \neg C)$</th>
<th>$KB$</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>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

CS 1571 Intro to AI
M. Hauskrecht
Truth-table approach

\[ KB = (A \lor C) \land (B \lor \neg C) \quad \alpha = (A \lor B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>( A \lor C )</th>
<th>( B \lor \neg C )</th>
<th>KB</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>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</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>True</td>
<td>False</td>
</tr>
</tbody>
</table>

KB entails \( \alpha \)

- 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?

- ?
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?
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:** check only entries for which KB is True.

This is the idea behind the **inference rules approach**

**Inference rules:**
- Represent sound inference patterns repeated in inferences
- Can be used to generate new (sound) sentences from the existing ones
Inference rules for logic

• **Modus ponens**

\[
A \Rightarrow B, \quad A \quad \Rightarrow \quad B
\]

• 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.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Inference rules for logic

• **And-elimination**

\[
A_1 \land A_2 \land \ldots \land A_n \quad \Rightarrow \quad A_i
\]

• **And-introduction**

\[
A_1, A_2, \ldots, A_n \quad \Rightarrow \quad A_1 \land A_2 \land \ldots \land A_n
\]

• **Or-introduction**

\[
A_i \quad \Rightarrow \quad A_1 \lor A_2 \lor \ldots \lor A_i \lor A_n
\]
Inference rules for logic

- **Elimination of double negation**
  \[ \neg\neg A \vdash A \]

- **Unit resolution**
  \[ A \lor B, \neg A \vdash B \]

- **Resolution**
  \[ A \lor B, \neg B \lor C \vdash A \lor C \]

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

1. \( P \land Q \)
2. \( P \Rightarrow R \)
3. \( (Q \land R) \Rightarrow S \)
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. \( P \) \hspace{1cm} *From 1 and And-elim*

\[
\frac{A_1 \land A_2 \land \ldots \land A_n}{A_i}
\]

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. \( P \)
5. \( R \) \hspace{1cm} *From 2,4 and Modus ponens*

\[
\frac{A \Rightarrow B, \quad A}{B}
\]
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. \( P \)
5. \( R \)
6. \( Q \)  
   From 1 and And-elim
   \[
   \frac{A_1 \land A_2 \land \ldots \land A_n}{A_i}
   \]

7. \( (Q \land R) \)  
   From 5,6 and And-introduction
   \[
   \frac{A_1, A_2, A_n}{A_1 \land A_2 \land \ldots \land A_n}
   \]
Example. Inference rules approach.

\[ \text{KB:} \ P \land Q \quad P \Rightarrow R \quad (Q \land R) \Rightarrow S \quad \text{Theorem:} \ S \]

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

\[ \text{Proved: } S \]

Example. Inference rules approach.

\[ \text{KB:} \ P \land Q \quad P \Rightarrow R \quad (Q \land R) \Rightarrow S \quad \text{Theorem:} \ S \]

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

\[ \text{Proved: } S \]
Inference rules

• To show that theorem $\alpha$ holds for a KB
  – we may need to apply a number of sound inference rules

  **Problem**: many possible inference rules to be applied next

  **Looks familiar?**

\[
\begin{align*}
P \Rightarrow Q \\
R \Rightarrow S \\
P \\
R \\
\ldots
\end{align*}
\]

Logic inferences and search

• To show that theorem $\alpha$ 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?**

\[
\begin{align*}
P \Rightarrow Q \\
P \\
R \Rightarrow S \\
S
\end{align*}
\]

**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 $\alpha$ 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

Normal forms

- **Problem**: Too many inference rules
- Can we simplify inferences using one of the normal forms?

Normal forms used:

**Conjunctive normal form (CNF)**
- conjunction of clauses (clauses include disjunctions of literals)
  \[(A \lor B) \land (\neg A \lor \neg C \lor D)\]

**Disjunctive normal form (DNF)**
- Disjunction of terms (terms include conjunction of literals)
  \[(A \land \neg B) \lor (\neg A \land C) \lor (C \land \neg D)\]
Conversion to a CNF

Assume: \( \neg(A \Rightarrow B) \lor (C \Rightarrow A) \)

1. Eliminate \( \Rightarrow, \iff \)
   \[ \neg(\neg A \lor B) \lor (\neg C \lor A) \]

2. Reduce the scope of signs through DeMorgan Laws and double negation
   \[ (A \land \neg B) \lor (\neg C \lor A) \]

3. Convert to CNF using the associative and distributive laws
   \[ (A \lor \neg C \lor A) \land (\neg B \lor \neg C \lor A) \]
   and
   \[ (A \lor \neg C) \land (\neg B \lor \neg C \lor A) \]

Inferences in CNF

Assume: \( \neg(A \Rightarrow B) \lor (C \Rightarrow A) \)

1. Eliminate \( \Rightarrow, \iff \)
   \[ \neg(\neg A \lor B) \lor (\neg C \lor A) \]

2. Reduce the scope of signs through DeMorgan Laws and double negation
   \[ (A \land \neg B) \lor (\neg C \lor A) \]

3. Convert to CNF using the associative and distributive laws
   \[ (A \lor \neg C \lor A) \land (\neg B \lor \neg C \lor A) \]
   and
   \[ (A \lor \neg C) \land (\neg B \lor \neg C \lor A) \]
Resolution rule

Resolution rule
• sound inference rule that fits the CNF

\[
\frac{A \lor B, \quad \neg B \lor C}{A \lor C}
\]

<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>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</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>False</td>
<td>True</td>
<td>False</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>False</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 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: \((A \land B)\)
  We want to show: \((A \lor B)\)

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)

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

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

\[ \text{M. Hauskrecht} \]
Inference problem and satisfiability

Inference problem:
• we want to show that the 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:
$$KB \models \alpha \text{ if and only if } (KB \land \neg \alpha) \text{ is unsatisfiable}$$

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 $\alpha$:
• **Incomplete**: 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 is incomplete

A trick to make things work:
• proof by contradiction
  – Disproving: $KB \land \neg \alpha$
  – 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:
    • $A, \neg A \rightarrow Q$
    • proves entailment.
  – No more new sentences can be derived
    • disproves it.

Example. Resolution.

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

Step 1. convert KB to CNF:
• $P \land Q$ $\longrightarrow$ $P \land Q$
• $P \Rightarrow R$ $\longrightarrow$ $(\neg P \lor R)$
• $(Q \land R) \Rightarrow S$ $\longrightarrow$ $(\neg Q \lor \neg R \lor S)$

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

Step 2. Negate the theorem to prove it via refutation
$S$ $\longrightarrow$ $\neg S$

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

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

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

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

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

\[
\begin{align*}
P & \quad Q & (\neg P \lor R) & (\neg Q \lor \neg R \lor S) & \neg S \\
\quad & R & (\neg R \lor S) & & \\
\quad & S & & & \\
\end{align*}
\]

Example. Resolution.

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

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

\[
\begin{align*}
P & \quad Q & (\neg P \lor R) & (\neg Q \lor \neg R \lor S) & \neg S \\
\quad & R & (\neg R \lor S) & & \\
\quad & S & & & \\
\end{align*}
\]

**Example. Resolution.**

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

\[
\begin{align*}
P & \quad Q & \quad (\neg P \lor R) & \quad (\neg Q \lor \neg R \lor S) & \quad \neg S \\
R & \quad & & (\neg R \lor S) & \\
 & & & S & \\
\end{align*}
\]

Contradiction \(\rightarrow \{\}\)  
**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