Propositional logic

Logic

A formal language for expressing knowledge and ways of reasoning.

Logic is defined by:

- **A set of sentences**
  - A sentence is constructed from a set of primitives according to syntax rules.

- **A set of interpretations**
  - An interpretation gives a semantic to primitives. It associates primitives with values.

- **The valuation (meaning) function** \( V \)
  - Assigns a value (typically the truth value) to a given sentence under some interpretation

\[
V : \text{sentence} \times \text{interpretation} \rightarrow \{ \text{True} , \text{False} \}
\]
Propositional logic. Syntax

• Formally propositional logic $P$:
  – Is defined by Syntax+interpretation+semantics of $P$

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:
    $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$

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
    \[ \neg A \quad (A \land B) \quad (A \lor B) \quad (A \Rightarrow B) \quad (A \Leftrightarrow B) \]
    or \[ (A \lor B) \land (A \lor \neg B) \]
    are sentences
Semantic: propositional symbols

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

\[ I: \text{Light in the room is on} \rightarrow \text{True}, \text{It rains outside} \rightarrow \text{False} \]

\[ V(\text{Light in the room is on}, I) = \text{?} \]

\[ V(\text{It rains outside}, I) = \text{?} \]

\[ V(\text{Light in the room is on} \land \text{It rains outside}, I) = \text{?} \]
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: $KB \vdash \alpha$

**Properties of the inference procedure in terms of entailment**
- **Soundness:** An inference procedure is sound
- **Completeness:** An inference procedure is complete
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**
  
  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

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 table:
- enumerates truth values of sentences for all possible interpretations (assignments of True/False to propositional symbols)

Example:

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

\( \alpha = (P \lor Q) \land Q \)
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 proposition symbols

\[ 2^n \] Rows in the table has to be filled

But typically only for a small subset of rows the KB is true

Problem with the truth table approach:

- the truth table is \textit{exponential} in the number of propositional symbols (we checked all assignments)
- KB is true on only a smaller subset
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)
- KB is true only on a small subset interpretations

**How to make the process more efficient?**

---

Inference rules 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)
- KB is true only on a smaller subset

**How to make the process more efficient?**

**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**

\[ \frac{A \Rightarrow B, \ A}{B} \]

- 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.

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( A \Rightarrow B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

- **And-elimination**

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

- **And-introduction**

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

- **Or-introduction**

\[ \frac{A_i}{A_1 \lor A_2 \lor \ldots \lor A_i \lor \ldots \lor A_n} \]
Inference rules for logic

- **Elimination of double negation**
  \[ \neg \neg A \quad \Rightarrow \quad A \]

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

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

- 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:** \( P \land Q \quad P \Rightarrow R \quad (Q \land R) \Rightarrow S \) \hspace{1cm} **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} \text{From 1 and And-elim}
   \[ \frac{A_1 \land A_2 \land A_n}{A_i} \]
5. \( R \)  \hspace{1cm} \text{From 2,4 and Modus ponens}
   \[ \frac{A \Rightarrow B, \ A}{B} \]
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

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

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. \( R \)
6. \( Q \)
7. \( (Q \land R) \)

From 5,6 and And-introduction

\[
\frac{A_1, A_2, A_n}{A_1 \land A_2 \land A_n}
\]
Example. Inference rules approach.

**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. $Q \quad A \Rightarrow B, \quad A \quad \frac{B}{A}$
7. $(Q \land R)$
8. $S$ \quad \text{From 7,3 and Modus ponens}

**Proved:** $S$

---

Example. Inference rules approach.

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

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

**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?**

\[
\begin{array}{c}
P \Rightarrow Q \\
R \Rightarrow S \\
P \\
R \\
\ldots
\end{array} \quad \begin{array}{c}
P \Rightarrow Q, \ P \\
Q \\
R \Rightarrow S, \ R \\
S
\end{array}
\]

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?**

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

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)
  
  $$(A \lor B) \land (\neg A \lor C \lor D)$$

**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 \( \Rightarrow, \Leftrightarrow \)
   \[
   \neg(\neg A \lor B) \lor (\neg C \lor A)
   \]

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
   \[
   (A \lor \neg C) \land (\neg B \lor \neg C \lor A)
   \]

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

- **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 $\alpha$ 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:**
- inference problem is NP-complete
- programs for solving the SAT problem can be used to solve the inference problem

---

Universal inference rule: Resolution rule

Sometimes inference rules can be combined into a single rule

**Resolution rule**
- sound inference rule that works for CNF
- It is complete for propositional logic (refutation complete)

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

<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>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</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>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
Universal rule: Resolution.

Initial obstacle:
• 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 ??)

A trick to make things work:
• proof by contradiction
  – Disproving:  \(KB, \neg \alpha\)
  – Proves the entailment  \(KB \models \alpha\)

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:
    • \(A, \neg A \rightarrow \Box\)
    • 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:
- \(P \land Q \quad \rightarrow \quad P \land Q\)
- \(P \Rightarrow R \quad \rightarrow \quad (\neg P \lor R)\)
- \((Q \land R) \Rightarrow S \quad \rightarrow \quad (\neg Q \lor \neg R \lor S)\)

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: \(S\)

\[
\begin{array}{c}
P \quad Q \\
\quad (\neg P \lor R) \\
\quad (\neg Q \lor \neg R \lor S) \\
\quad \neg S
\end{array}
\]

\[
\begin{array}{c}
R
\end{array}
\]

Example. Resolution.

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

\[
\begin{array}{c}
P \quad Q \\
\quad (\neg P \lor R) \\
\quad (\neg Q \lor \neg R \lor S) \\
\quad \neg S
\end{array}
\]

\[
\begin{array}{c}
R \\
\quad (\neg R \lor S)
\end{array}
\]
Example. Resolution.

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

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

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

\[ S \]

Contradiction $\rightarrow \{\}$

Proved: $S$