Propositional logic

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

Knowledge representation
Knowledge-based agent

- **Knowledge base (KB):**
  - A set of sentences that describe facts about the world in some formal (representational) language
  - **Domain specific**
- **Inference engine:**
  - A set of procedures that use the representational language to infer new facts from known ones or answer a variety of KB queries. Inferences typically require search.
  - **Domain independent**

Example: MYCIN

- MYCIN: an expert system for diagnosis of bacterial infections
- **Knowledge base** represents
  - Facts about a specific patient case
  - Rules describing relations between entities in the bacterial infection domain

\[
\begin{align*}
\text{If} & \quad 1. \text{The stain of the organism is gram-positive, and} \\
& \quad 2. \text{The morphology of the organism is coccus, and} \\
& \quad 3. \text{The growth conformation of the organism is chains} \\
\text{Then} & \quad \text{the identity of the organism is streptococcus}
\end{align*}
\]

- **Inference engine:**
  - manipulates the facts and known relations to answer diagnostic queries (consistent with findings and rules)
Knowledge representation

- The objective of knowledge representation is to express the knowledge about the world in a computer-tractable form.

- Key aspects of knowledge representation languages:
  - **Syntax**: describes how sentences are formed in the language.
  - **Semantics**: describes the meaning of sentences, what is it the sentence refers to in the real world.
  - **Computational aspect**: describes how sentences and objects are manipulated in concordance with semantical conventions.

Many KB systems rely on some variant of 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.

$$V : \text{sentence} \times \text{interpretation} \rightarrow \{\text{True}, \text{False}\}$$
Example of logic

Language of numerical constraints:

• A sentence:
  \[ x + 3 \leq z \]
  \( x, z \) - variable symbols (primitives in the language)

• An interpretation:
  I: \( x = 5, z = 2 \)
  Variables mapped to specific real numbers

• Valuation (meaning) function \( V \):
  \[ V ( x + 3 \leq z, I ) \] is \textit{False} for I: \( x = 5, z = 2 \)
  is \textit{True} for I: \( x = 5, z = 10 \)

Types of logic

• Different types of logics possible:
  – Propositional logic
  – First-order logic
  – Temporal logic
  – Numerical constraints logic
  – Map-coloring logic

In the following:

• **Propositional logic.**
  – Formal language for making logical inferences
  – Foundations of \textit{propositional logic: George Boole} (1854)
Propositional logic. Syntax

- **Propositional logic P:**
  - defines a language for symbolic reasoning

- **Proposition:** a statement that is either true or false

- **Examples of propositions:**
  - *Pitt is located in the Oakland section of Pittsburgh.*
  - *France is in Europe.*
  - *It rains outside.*
  - *2 is a prime number and 6 is a prime*
  - *How are you?* Not a proposition.

---

**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:**
    - $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$
Propositional logic. Syntax

Sentences in the propositional logic:

- **Atomic sentences:**
  - Constructed from constants and propositional symbols
  - True, False are (atomic) sentences
  - \( P \cdot Q \) 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
    \(-A, (A \land B), (A \lor B), (A \Rightarrow B), (A \Leftrightarrow B)\)
    or \((A \lor B) \land (A \lor \neg B)\)
  - 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
• 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 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

\[ V(\text{Light in the room is on, } \mathbf{I}) = \text{True} \]
\[ V(\text{It rains outside, } \mathbf{I}) = \text{False} \]

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

\[ V(\text{Light in the room is on, } \mathbf{I’}) = \text{False} \]
Semantics: constants

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

\[
V(\text{True, } I) = True \\
V(\text{False, } I) = False
\]

For any interpretation $I$

Semantics: composite sentences.

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

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\neg P$</th>
<th>$P \land Q$</th>
<th>$P \lor Q$</th>
<th>$P \Rightarrow Q$</th>
<th>$P \Leftarrow Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>False</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>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</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>
<td>True</td>
</tr>
</tbody>
</table>
Translation

Assume the following sentences:

• It is not sunny this afternoon and it is colder than yesterday. \( \neg p \land q \)
• We will go swimming only if it is sunny.
• If we do not go swimming then we will take a canoe trip.
• If we take a canoe trip, then we will be home by sunset.

Denote:

• \( p \): It is sunny this afternoon
• \( q \): it is colder than yesterday
• \( r \): We will go swimming
• \( s \): we will take a canoe trip
• \( t \): We will be home by sunset
Translation

Assume the following sentences:
• It is not sunny this afternoon and it is colder than yesterday. \( \neg p \land q \)
• We will go swimming only if it is sunny. \( r \rightarrow p \)
• If we do not go swimming then we will take a canoe trip. \( \neg r \rightarrow s \)
• If we take a canoe trip, then we will be home by sunset.

Denote:
• \( p \) = It is sunny this afternoon
• \( q \) = it is colder than yesterday
• \( r \) = We will go swimming
• \( s \) = we will take a canoe trip
• \( t \) = We will be home by sunset
Translation

Assume the following sentences:
• It is not sunny this afternoon and it is colder than yesterday.  \( \neg p \land q \)
• We will go swimming only if it is sunny.  \( r \rightarrow p \)
• If we do not go swimming then we will take a canoe trip.  \( \neg r \rightarrow s \)
• If we take a canoe trip, then we will be home by sunset.  \( s \rightarrow t \)

Denote:
• \( p \) = It is sunny this afternoon
• \( q \) = it is colder than yesterday
• \( r \) = We will go swimming
• \( s \) = we will take a canoe trip
• \( t \) = We will be home by sunset

Contradiction and Tautology

Some composite sentences may always (under any interpretation) evaluate to a single truth value:
• **Contradiction** (always False)
  \[ P \land \neg P \]
• **Tautology** (always True)
  \[ P \lor \neg P \]

\[
\neg (P \lor Q) \iff (\neg P \land \neg Q) \\
\neg (P \land Q) \iff (\neg P \lor \neg Q)
\]

\( \text{DeMorgan’s Laws} \)
Model, validity and satisfiability

- A **model (in logic)**: An interpretation is a model for a set of sentences if it assigns true to each sentence in the set.

- A sentence is **satisfiable** if it has a model;
  - There is at least one interpretation under which the sentence can evaluate to True.

- A sentence is **valid** if it is *True* in all interpretations
  - i.e., if its negation is **not satisfiable** (leads to contradiction)

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

CS 2710 Foundations of AI
Model, validity and satisfiability

- A **model (in logic)**: An interpretation is a model for a set of sentences if it assigns true to each sentence in the set.
- A sentence is **satisfiable** if it has a model;
  - There is at least one interpretation under which the sentence can evaluate to True.
- A sentence is **valid** if it is *True* in all interpretations
  - i.e., if its negation is **not satisfiable** (leads to contradiction)

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

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \lor Q$</th>
<th>$(P \lor Q) \land \neg Q$</th>
<th>$((P \lor Q) \land \neg Q) \Rightarrow P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</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>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>
Entailment

- **Entailment** reflects the relation of one fact in the world following from the others

- Entailment $KB \models \alpha$
- Knowledge base KB entails sentence $\alpha$ if and only if $\alpha$ is true in all worlds where KB is 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: $KB \models_i \alpha$

**Properties of the inference procedure in terms of entailment**
- **Soundness:** An inference procedure is **sound**
  
  If $KB \models_i \alpha$ then it is true that $KB \models \alpha$

- **Completeness:** An inference procedure is **complete**
  
  If $KB \models \alpha$ then it is true that $KB \models_i \alpha$
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?

**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**.

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 values to propositional symbols)

Example:

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

CS 2710 Foundations of AI
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>( P )</th>
<th>( Q )</th>
<th>( P \lor Q )</th>
<th>( P \iff Q )</th>
<th>( (P \lor \neg Q) \land Q )</th>
</tr>
</thead>
<tbody>
<tr>
<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>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</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>False</td>
<td>False</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>True</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>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</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>True</td>
<td>True</td>
<td>True</td>
<td>True</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>
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: $KB = (A \lor C) \land (B \lor \lnot C)$, $\alpha = (A \lor B)$

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

CS 2710 Foundations of AI
Truth-table approach

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

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>A \lor C</td>
<td>(B \lor \neg C)</td>
<td>KB</td>
<td>\alpha</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---------</td>
<td>-----------</td>
<td>----</td>
<td>------</td>
</tr>
<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>False</td>
<td>True</td>
<td>False</td>
<td>True</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>True</td>
<td>False</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 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
Limitations 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 on only a smaller subset
Inference rules approach.

\[ KB \models \alpha \, ? \]

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.

How to make the process more efficient?

Solution: check only entries for which KB is \textit{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

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

\[
B \quad \quad \quad \text{conclusion}
\]

- If both sentences in the premise are true then conclusion is true.
- The modus ponens inference rule is \textit{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>
Inference rules for logic

• And-elimination

\[ A_1 \wedge A_2 \wedge A_n \quad A_i \]

• And-introduction

\[ A_1, A_2, A_n \quad A_1 \wedge A_2 \wedge A_n \]

• Or-introduction

\[ A_i \quad A_1 \vee A_2 \vee \ldots A_i \vee A_n \]

Inference rules for logic

• Elimination of double negation

\[ \neg\neg A \quad A \]

• Unit resolution

\[ A \vee B, \neg A \quad B \]

• Resolution

\[ A \vee B, \neg B \vee C \quad A \vee C \]

• 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:** $P \land Q$  \hspace{0.5cm} $P \Rightarrow R$  \hspace{0.5cm} $(Q \land R) \Rightarrow S$  \hspace{0.5cm} **Theorem:** $S$

1. $P \land Q$
2. $P \Rightarrow R$
3. $(Q \land R) \Rightarrow S$

---

Example. Inference rules approach.

**KB:** $P \land Q$  \hspace{0.5cm} $P \Rightarrow R$  \hspace{0.5cm} $(Q \land R) \Rightarrow S$  \hspace{0.5cm} **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:** \( P \land Q \quad P \Rightarrow R \quad (Q \land R) \Rightarrow S \quad \text{Theorem:} \ S \)

1. \( P \land Q \)
2. \( P \Rightarrow R \)
3. \( (Q \land R) \Rightarrow S \)
4. \( P \)
5. \( R \quad \text{From } 2, 4 \text{ and Modus ponens} \)
   \[
   \frac{A \Rightarrow B, \ A}{B}
   \]

6. \( Q \quad \text{From } 1 \text{ and And-elim} \)
   \[
   \frac{A_1 \land A_2 \land \ldots \land A_n}{A_i}
   \]
Example. Inference rules approach.

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

1. $P \land Q$
2. $P \Rightarrow R$
3. $(Q \land R) \Rightarrow S$
4. $P$
5. $R$
6. $Q$
7. $(Q \land R)$  \hspace{1cm} \text{From 5,6 and And-introduction}
   \hspace{1cm} \frac{A_1, A_2, \ldots, A_n}{A_1 \land A_2 \land \cdots \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. $Q$
7. $(Q \land R)$  \hspace{1cm} \text{From 7,3 and Modus ponens}
8. $S$ \hspace{1cm} \text{Proved: } S
Example. Inference rules approach.

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

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

---

**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?**

---

CS 2710 Foundations of AI
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

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