Chapter 3: Methods of Inference


Objectives

- Learn the definitions of trees, lattices, and graphs
- Learn about state and problem spaces
- Learn about AND-OR trees and goals
- Explore different methods and rules of inference
- Learn the characteristics of first-order predicate logic and logic systems
Objectives

- Discuss the resolution rule of inference, resolution systems, and deduction
- Compare shallow and causal reasoning
- How to apply resolution to first-order predicate logic
- Learn the meaning of forward and backward chaining

Objectives

- Explore additional methods of inference
- Learn the meaning of Metaknowledge
- Explore the Markov decision process
Trees

- A tree is a hierarchical data structure consisting of:
  - Nodes – store information
  - Branches – connect the nodes
- The top node is the root, occupying the highest hierarchy.
- The leaves are at the bottom, occupying the lowest hierarchy.

Every node, except the root, has exactly one parent.

Every node may give rise to zero or more child nodes.

A binary tree restricts the number of children per node to a maximum of two.

Degenerate trees have only a single pathway from root to its one leaf.
Graphs

- Graphs are sometimes called a network or net.
- A graph can have zero or more links between nodes – there is no distinction between parent and child.
- Sometimes links have weights – weighted graph; or, arrows – directed graph.
- Simple graphs have no loops – links that come back onto the node itself.
Graphs

- A circuit (cycle) is a path through the graph beginning and ending with the same node.
- Acyclic graphs have no cycles.
- Connected graphs have links to all the nodes.
- Digraphs are graphs with directed links.
- Lattice is a directed acyclic graph.

**Figure 3.2 Simple Graphs**

(a) A nonconnected graph  
(b) A connected graph  
(c) A digraph with a self-loop and circuit  
(d) A lattice  
(e) Degenerate binary trees of three nodes
Making Decisions

- Trees / lattices are useful for classifying objects in a hierarchical nature.
- Trees / lattices are useful for making decisions.
- We refer to trees / lattices as structures.
- Decision trees are useful for representing and reasoning about knowledge.

Binary Decision Trees

- Every question takes us down one level in the tree.
- A binary decision tree having $N$ nodes:
  - All leaves will be answers.
  - All internal nodes are questions.
  - There will be a maximum of $2^N$ answers for $N$ questions.
- Decision trees can be self learning.
- Decision trees can be translated into production rules.
Decision Tree Example

State and Problem Spaces

- A state space can be used to define an object’s behavior.
- Different states refer to characteristics that define the status of the object.
- A state space shows the transitions an object can make in going from one state to another.
Finite State Machine

- A FSM is a diagram describing the finite number of states of a machine.
- At any one time, the machine is in one particular state.
- The machine accepts input and progresses to the next state.
- FSMs are often used in compilers and validity checking programs.

Using FSM to Solve Problems

- Characterizing ill-structured problems – one having uncertainties.
- Well-formed problems:
  - Explicit problem, goal, and operations are known
  - Deterministic – we are sure of the next state when an operator is applied to a state.
  - The problem space is bounded.
  - The states are discrete.
AND-OR Trees and Goals

- 1990s, PROLOG was used for commercial applications in business and industry.
- PROLOG uses backward chaining to divide problems into smaller problems and then solves them.
- AND-OR trees also use backward chaining.
- AND-OR-NOT lattices use logic gates to describe problems.
Types of Logic

- **Deduction** – reasoning where conclusions must follow from premises
- **Induction** – inference is from the specific case to the general
- **Intuition** – no proven theory
- **Heuristics** – rules of thumb based on experience
- **Generate and test** – trial and error

Types of Logic

- **Abduction** – reasoning back from a true condition to the premises that may have caused the condition
- **Default** – absence of specific knowledge
- **Autoepistemic** – self-knowledge
- **Nonmonotonic** – previous knowledge
- **Analogy** – inferring conclusions based on similarities with other situations
Deductive Logic

• Argument – group of statements where the last is justified on the basis of the previous ones

• Deductive logic can determine the validity of an argument.

• Syllogism – has two premises and one conclusion

• Deductive argument – conclusions reached by following true premises must themselves be true

Syllogisms vs. Rules

• Syllogism:
  – All basketball players are tall.
  – Jason is a basketball player.
  – Jason is tall.

• IF-THEN rule:
  IF All basketball players are tall and Jason is a basketball player
  THEN Jason is tall.
Categorical Syllogism

Premises and conclusions are defined using categorical statements of the form:

Table 3.2 Categorical Statements

<table>
<thead>
<tr>
<th>Form</th>
<th>Schema</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>All S is P</td>
<td>universal affirmative</td>
</tr>
<tr>
<td>E</td>
<td>No S is P</td>
<td>universal negative</td>
</tr>
<tr>
<td>I</td>
<td>Some S is P</td>
<td>particular affirmative</td>
</tr>
<tr>
<td>O</td>
<td>Some S is not P</td>
<td>particular negative</td>
</tr>
</tbody>
</table>

Categorical Syllogisms

middle term – common to both premises

All A is B
All C is A

All C is B

subject of the conclusion is called the major term

C – the subject of the conclusion is called the minor term
Categorical Syllogisms

To write the syllogism in standard form, we would write:

Major Premise: All A is B
Minor Premise: All C is A

Conclusion: All C is B

Proving the Validity of Syllogistic Arguments Using Venn Diagrams

1. If a class is empty, it is shaded.
2. Universal statements, A and E are always drawn before particular ones.
3. If a class has at least one member, mark it with an *.
4. If a statement does not specify in which of two adjacent classes an object exists, place an * on the line between the classes.
5. If an area has been shaded, not * can be put in it.
Rules of Inference

• Venn diagrams are insufficient for complex arguments.

• Syllogisms address only a small portion of the possible logical statements.

• Propositional logic offers another means of describing arguments.

Direct Reasoning

Modus Ponens

If I am hungry, I will eat
I am hungry

\[ \therefore \text{ I will eat} \]

\[ A = \text{ I am hungry} \]

\[ B = \text{ I will eat} \]

\[ A \rightarrow B \]

\[ \therefore B \]
Truth Table Modus Ponens

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p → q</th>
<th>(p → q) ∨ p</th>
<th>(p → q) ∧ p → q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

Some Rules of Inference

1. Law of Detachment
   \[ p → q \]
   \[ p \]
   \[ ∴ q \]

2. Law of the Contrapositive
   \[ p → q \]
   \[ ∴ -q → -p \]

3. Law of Modus Tollens
   \[ p → q \]
   \[ -q \]
   \[ ∴ -p \]

4. Chain Rule (Law of the Syllogism)
   \[ p → q \]
   \[ q → r \]
   \[ ∴ p → r \]

5. Law of Disjunctive Inference
   \[ p ∨ q \]
   \[ ¬p \]
   \[ ∴ q \]
   \[ p ∨ q \]
   \[ ¬q \]
   \[ ∴ p \]
Rules of Inference

\[ \neg \neg p \quad \therefore p \]

7. De Morgan's Law
\[ \neg (p \land q) \quad \therefore \neg p \lor \neg q \]
\[ \neg (p \lor q) \quad \therefore \neg p \land \neg q \]

8. Law of Simplification
\[ p \land q \quad \therefore p \]
\[ p \land q \quad \therefore q \]

9. Law of Conjunction
\[ p \quad q \quad \therefore p \land q \]
\[ p \quad \therefore p \lor q \]

10. Law of Disjunctive Addition
\[ p \quad \therefore \neg (p \land q) \]
\[ q \quad \therefore \neg (p \land q) \]

11. Law of Conjunctive Argument
\[ \neg q \quad \therefore \neg p \]

Table 3.9 The Modus Meanings

<table>
<thead>
<tr>
<th>Rule of Inference Number</th>
<th>Name</th>
<th>Meaning “mood which by…”</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>modus ponendo ponens</td>
<td>affirming, affirms</td>
</tr>
<tr>
<td>3</td>
<td>modus tollendo tollens</td>
<td>denying, denies</td>
</tr>
<tr>
<td>5</td>
<td>modus tollendo ponens</td>
<td>denying, affirms</td>
</tr>
<tr>
<td>11</td>
<td>modus ponendo tollens</td>
<td>affirming, denies</td>
</tr>
</tbody>
</table>
Table 3.10 The Conditional and Its Variants

<table>
<thead>
<tr>
<th></th>
<th>conditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>converse</td>
<td>p → q</td>
</tr>
<tr>
<td>inverse</td>
<td>p → q</td>
</tr>
<tr>
<td>contrapositive</td>
<td>q → p</td>
</tr>
</tbody>
</table>

Limitations of Propositional Logic

- If an argument is invalid, it should be interpreted as such – that the conclusion is necessarily incorrect.
- An argument may be invalid because it is poorly concocted.
- An argument may not be provable using propositional logic, but may be provable using predicate logic.
First-Order Predicate Logic

- Syllogistic logic can be completely described by predicate logic.
- The Rule of Universal Instantiation states that an individual may be substituted for a universe.

Logic Systems

- A logic system is a collection of objects such as rules, axioms, statements, and so forth in a consistent manner.
- Each logic system relies on formal definitions of its axioms (postulates) which make up the formal definition of the system.
- Axioms cannot be proven from within the system.
- From axioms, it can be determined what can be proven.
Goals of a Logic System

• Be able to specify the forms of arguments – well formulated formulas – wffs.

• Indicate the rules of inference that are invalid.

• Extend itself by discovering new rules of inference that are valid, extending the range of arguments that can be proven – theorems.

Requirements of a Formal System

1. An alphabet of symbols
2. A set of finite strings of these symbols, the wffs.
3. Axioms, the definitions of the system.
4. Rules of inference, which enable a wff to be deduced as the conclusion of a finite set of other wffs – axioms or other theorems of the logic system.
Requirements of a FS Continued

5. Completeness – every wff can either be proved or refuted.
6. The system must be sound – every theorem is a logically valid wff.

Shallow and Causal Reasoning

• Experiential knowledge is based on experience.
• In shallow reasoning, there is little/no causal chain of cause and effect from one rule to another.
• Advantage of shallow reasoning is ease of programming.
• Frames are used for causal / deep reasoning.
• Causal reasoning can be used to construct a model that behaves like the real system.
Converting First-Order Predicate wffs to Clausal Form

1. Eliminate conditionals.
2. When possible, eliminate negations or reduce their scope.
3. Standardize variables.
4. Eliminate existential quantifiers using Skolem functions.
5. Convert wff to prenex form.

Converting

6. Convert the matrix to conjunctive normal form.
7. Drop the universal quantifiers as necessary.
8. Eliminate \( \lor \) signs by writing the wff as a set of clauses.
9. Rename variables in clauses making unique.
Chaining

• Chain – a group of multiple inferences that connect a problem with its solution
• A chain that is searched / traversed from a problem to its solution is called a forward chain.
• A chain traversed from a hypothesis back to the facts that support the hypothesis is a backward chain.
• Problem with backward chaining is find a chain linking the evidence to the hypothesis.

Figure 3.21 Causal Forward Chaining
Table 3.14 Some Characteristics of Forward and Backward Chaining

<table>
<thead>
<tr>
<th>Forward Chaining</th>
<th>Backward Chaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning, monitoring, control</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Present to future</td>
<td>Present to past</td>
</tr>
<tr>
<td>Antecedent to consequent</td>
<td>Consequent to antecedent</td>
</tr>
<tr>
<td>Data driven, bottom-up reasoning</td>
<td>Goal driven, top-down reasoning</td>
</tr>
<tr>
<td>Work forward to find what solutions</td>
<td>Work backward to find facts</td>
</tr>
<tr>
<td>follow from the facts</td>
<td>that support the hypothesis</td>
</tr>
<tr>
<td>Breadth-first search facilitated</td>
<td>Depth-first search facilitated</td>
</tr>
<tr>
<td>Antecedents determine search</td>
<td>Consequences determine search</td>
</tr>
<tr>
<td>Explanation not facilitated</td>
<td>Explanation facilitated</td>
</tr>
</tbody>
</table>

Other Inference Methods

- Analogy – relating old situations (as a guide) to new ones.
- Generate-and-Test – generation of a likely solution then test to see if proposed meets all requirements.
- Abduction – Fallacy of the Converse
- Nonmonotonic Reasoning – theorems may not increase as the number of axioms increase.
Figure 3.14 Types of Inference

Metaknowledge

- The Markov decision process (MDP) is a good application to path planning.
- In the real world, there is always uncertainty, and pure logic is not a good guide when there is uncertainty.
- A MDP is more realistic in the cases where there is partial or hidden information about the state and parameters, and the need for planning.
Summary

• We have discussed the commonly used methods for inference for expert systems.
• Expert systems use inference to solve problems.
• We discussed applications of trees, graphs, and lattices for representing knowledge.
• Deductive logic, propositional, and first-order predicate logic were discussed.
• Truth tables were discussed as a means of proving theorems and statements.

Summary

• Characteristics of logic systems were discussed.
• Resolution as a means of proving theorems in propositional and first-order predicate logic.
• The nine steps to convert a wff to clausal form were covered.